
6,– DM
Ôs 50,–
sfr 6,–
Das führende Schneider-Magazin
CPC
Schneider
INTERNATIONAL
6
Juni 1986
2. Jahrgang
Aktuell:
MONTE-CARLO-SIMULATION
Im Test:
VORTEX FI-X
Joyce:
DATENÜBERTRAGUNG
MIT APPLE
LITBOX
Tips & Tricks:
SEIKOSHA-HARDCOPY
BASIC-TUNING
8K MEHR SPEICHER
Hardware:
CPC AM TV
EPROMMER
Viva Mexico - das
WM-Programm

Impressum

Schneider CPC International
erscheint in der
DMV - Daten & Medien Verlagsges. mbH

Chefredakteur
Christian Widuch (verantwortlich)

Stellvertretender Chefredakteur
Stefan Ritter

Redaktion
Thomas Morgen (TM), Michael Ebbrecht (ME), Heinrich Stiller (HS), Michael Ceol (MC)

Satz
Silvia Führer, Renate Wells

Gestaltung
Renate Wells, Gerd Köberich

Grafik/Illustration
Heinrich Stiller

Fotografie
Gerd Köberich

Anzeigenleitung
Wolfgang Schnell
Es gilt die Anzeigenpreisliste Nr. 2 vom 1.1.1986

Freie Mitarbeiter
Horst Franke (HF), S. Bokler, W. Fußnagel, Dipl.-Ing. H. Bruhns, M. Keuser, T. Kirsch, Matthias Uphoff, S. Koltmann, Dipl.-Ing. H. Scheruhn, R. Nitsche, T. Fippl, W. Wantia, J. Hüpper, F. Oppermann, Dipl.-Ing. D. Schröder, M. Braun, K. Wiltsch, Dr. M. Wolter

Anschrift Verlag/Redaktion
DMV - Daten & Medien Verlagsges. mbH
Postfach 250, Fuldaer Str. 6
3440 Eschwege
Telefon: 05651/8702

Vertrieb
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Vertrieb Österreich
Pressegroßvertrieb
Salzburg Ges.mbH & Co. KG
Niederalm 300
5081 Anif
Tel.: 06246/3721

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise
"Schneider CPC International" erscheint monatlich am Ende des Vormonats.
Einzelpreis DM 6,-/sfr. 6,-/ÖS 50,-

Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland:
Jahresbezugspreis: DM 60,-
Halbjahresbezugspreis: DM 30,-
Europäisches Ausland:
Jahresbezugspreis: DM 90,-
Halbjahresbezugspreis: DM 45,-
Außereuropäisches Ausland:
Jahresbezugspreis: DM 120,-
Halbjahresbezugspreis: DM 60,-
Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr. 23043-608
Raiffeisenbank Eschwege:
BLZ: 522 603 85, Kto.-Nr. 245 7008
Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim Verlag schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Monate, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird.



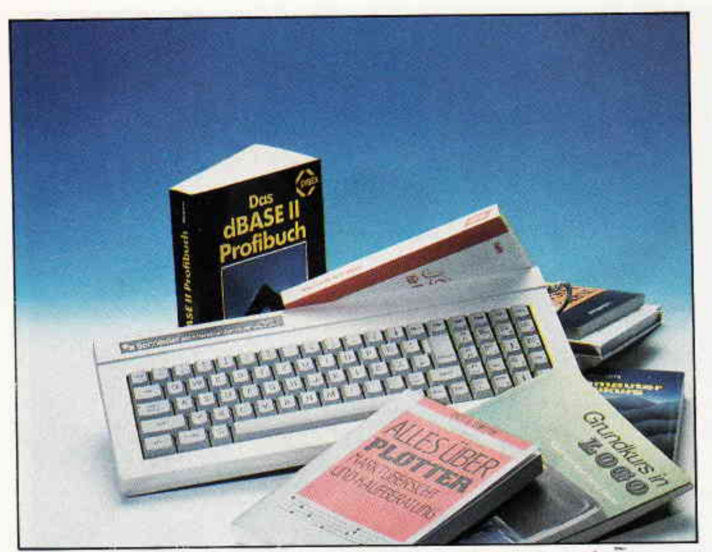


Liebe Leser,

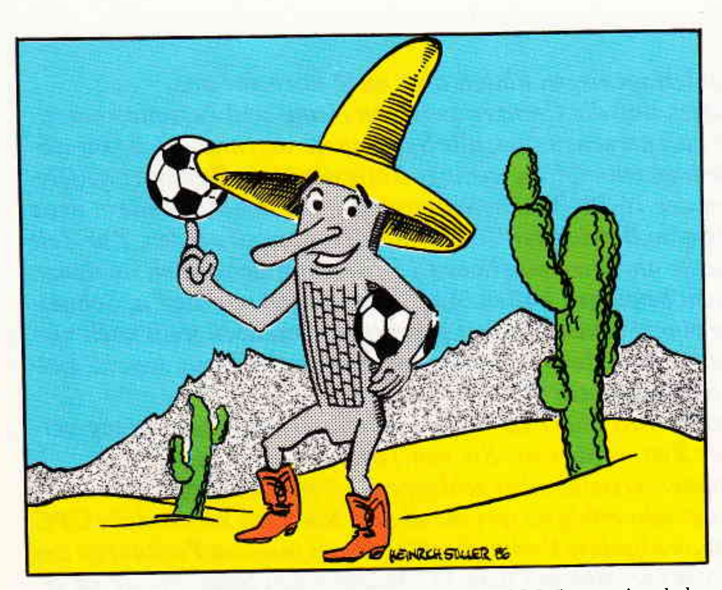
ereignisreiche Computertage liegen hinter, aber auch noch vor uns.
Allgemein wird erwartet, daß die Grenzen zwischen Heim- und Personalcomputern nur noch schwer auszumachen sind; alle Systeme werden immer leistungsfähiger, aber nicht unbedingt teurer. Eine wichtige Rolle werden in Zukunft zweifelsohne die Prozessoren der 16-Bit-Technologie spielen, die den Anwendern neue Perspektiven, vor allem im Bereich der Geschwindigkeit, bieten werden. Zur Zeit kursieren viele Gerüchte um einen möglichen 16-Biter von Schneider, an denen wir uns allerdings nicht beteiligen wollen. Auf Anfrage teilte uns die Fa. Schneider mit, daß zwar an einem neuen Rechner gearbeitet wird, doch weiß noch niemand, wie das Gerät im Endeffekt aussehen soll, geschweige denn, welche Leistung zu welchem Preis angeboten wird.
Ihre CPC International wird daher auf eine spekulative Berichterstattung verzichten. Zu gegebener Zeit werden wir Sie von Tatsachen unterrichten.
Während viele Computerhersteller über schleppende Verkäufe klagen, sieht die Situation bei Schneider sehr erfolgversprechend aus. Nach wie vor sind die CPC 464 und 6128-Computer absolute Verkaufsschlager. Als jüngstes Produkt ist der Joyce auf dem besten Wege, sich auf dem PC-Markt einen Spitzenplatz zu erobern. Wir haben dies zum Anlaß genommen, unsere Joyce-Rubrik zu erweitern, so daß auch die vielen interessierten Joyce-Besitzer entsprechende Informationen, Tips und Programme für den Umgang mit Ihrem Rechner erhalten.
In diesem Zusammenhang möchten wir noch einmal auf unseren Databox-Service hinweisen, der inzwischen um den Joyce erweitert wurde und zusätzliche Programme in Form von Spielen enthält.
Einen guten Griff haben wir mit unserer Hardware-Ecke getan. Die große Resonanz über den Auftaktartikel hat uns überwältigt.
Die konsequente Fortsetzung dieser Artikelserie beginnt in diesem Heft mit einem Grundlagenartikel über den Anschluß von Peripherie und einer kompletten Bauanleitung für einen Eprommer. Aus aktuellem Anlaß, die Fußball Weltmeisterschaft steht vor der Tür, haben wir unseren Programmservice auf dieses Ereignis abgestimmt und ein WM-Statistikprogramm für Sie ausgesucht. Bleibt zu hoffen, daß dieser Wettkampf die Völkerverständigung ähnlich erfolgreich fördert, wie wir das vom "Computern" auch über die Landesgrenzen hinaus gewohnt sind.

Herzlichst Ihr

Christian Widuch
Chefredakteur



Litbox - die Literaturkartei für den Joyce. Des weiteren gibt es viele Tips, Informationen und Software Reviews.



Rechtzeitig zur Fußball-WM in Mexico präsentiert CPC International das WM-Statistikprogramm. Damit Sie immer auf dem Laufenden sind!



Bild und Ton über TV – jetzt kein Problem mehr. Besonders interessant für die CPC-Besitzer mit Grünmonitor!

Berichte:

Serie:

Hardware:

Tips + Tricks:

CP/M:

Programme:

Lehrgänge:

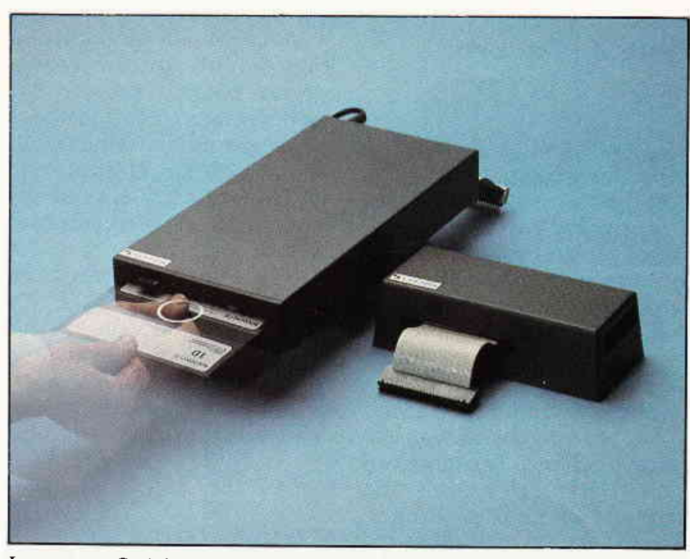
Joyce:

Referenzkarte:

Abenteuer:

Sofware Reviews:

Rubriken:



Interessante Peripherie von Vortex. Das neue 5 ¼"-Laufwerk F1-X mit hervorragenden Leistungsdaten.



Interessante Perspektiven eröffnen sich bei der Monte-Carlo-Simulation in Logo. Läßt sich der Zufall berechnen?



Crafton & Xunx ist nur ein Beispiel für die phantastische neue Software aus Frankreich.

Eine Bitte an unsere Leser

Die Rubrik »Leserbriefe« ist eine Einrichtung für alle Leser, die in irgendeiner Form Fragen, Probleme oder Anregungen zu Produkten, Programmierproblemen oder zu unserer Zeitschrift haben. Selbstverständlich sind wir bemüht, alle Leserfragen zu beantworten. Doch haben Sie bitte Verständnis, daß wir nicht alle eingehenden Briefe persönlich beantworten können. Oft erreichen uns mehrere Briefe zum gleichen Thema, einer davon wird dann stellvertretend für alle in unserer Zeitschrift beantwortet. Da auch wir nicht alle Fragen auf Anhieb beantworten können, müssen wir recherchieren. Und das dauert bekanntlich seine Zeit!! Wir möchten hiermit alle CPC-Leser noch einmal auf unseren Leser-Service hinweisen und bitten bei der Vielzahl der eingehenden Briefe um etwas Geduld. Für eilige Anfragen steht unsere Redaktion jeden Mittwoch von 17 – 20 Uhr am »Heißen Draht« zur Verfügung. Vielen Dank für Ihr Verständnis.

Ihre CPC-Redaktion

PS: Die Redaktion behält sich vor, Leserzuschriften in gekürzter Form wiederzugeben.

Anmerkung der Redaktion

Uns erreichten viele Zuschriften, die den großen Bereich der Computerspiele betreffen. Oftmals beklagen sich die Leser über den geringen Anteil von Software Reviews, Tips und Lösungen zu Adventures in der CPC International.

Wir sind selbstverständlich bemüht, für jeden Einsatzbereich der Schneider Computer entsprechende Informationen zu liefern und niemanden "zu kurz" kommen zu lassen.

Daß dies nicht immer zu 100 % realisiert werden kann, liegt oft am aktuellen Tagesgeschehen der Computerbranche und macht es notwendig, den einen oder anderen fest geplanten Beitrag aus eben genannten Gründen vorerst zu verschieben.

Als Bonbon für alle Spielefreaks und -interessierte wird die Juli-Ausgabe von CPC International dem Thema "Computerspiel" einen großen Sonderteil widmen.

Freuen Sie sich also auf die nächste CPC International, der 20 Seiten umfassende Spiele-Teil hat's in sich!

Do not disturb!

Zuerst möchte ich Ihnen zu Ihrer hervorragenden Zeitschrift "Schneider CPC International" gratulieren. Vor allem die Kurse über Computersprachen finde ich hochinteressant, und über den angekündigten Pascal-Kurs freue ich mich schon. Nur der Assemblerkurs ist stellenweise unverständlich. So gibt es z.B. wesentlich einfachere Algorithmen zur Multiplikation! Zur Assemblerprogrammierung habe ich auch noch eine Frage: Was gibt es bei der Benutzung des Befehls "EXX" zu beachten?

Mein CPC 464 stürzt jedesmal ab oder verzerrt das Schriftbild, was sich mit einem Moduswechsel wieder korrigieren läßt.

Alexander Hartmaier
Kirchenarnbach

CPC:

Der Assemblerbefehl "EXX" vertauscht den Standardregistersatz mit den Alternativregistern. In diesem Zustand hat jede Unterbrechung der Routine verheerende Folgen.

Sperren Sie vor der Anwendung alle Interrupts mit "DI" und geben Sie sie nach Abarbeitung und Zurückstellen der Register mit "EI" wieder frei. Dann kann die Routine nicht mehr "gestört" werden

Context 2

Ich bin seit vier Monaten Besitzer eines Seikosha SP-1000 CPC und habe mich deshalb unheimlich gefreut, durch Eure Zeitschrift an ein so starkes Textverarbeitungsprogramm zu kommen. Der Drucker Seikosha SP-1000 CPC verfügt serienmäßig über einen Schönschriftschalter. Ich kann also auf eine Einstellung über die Software verzichten. Die Speicherstellen &9F1A – &9F1C werden also frei und können anders genutzt werden.

Der Seikosha stellt als Schriftarten PICA und ELITE zur Verfügung. Es liegt also nahe, die freien Speicherstellen zu nutzen, zumal nur geringe Anpassungen im BASIC und einige POKEs notwendig sind.

Außerdem nutze ich den Drucker sehr viel zur Einzelblattverarbeitung. Ich finde es lästig, den Drucker ständig ein- und auszuschalten, um den komfortablen Papiereinzug wieder zu aktivieren. Deshalb habe ich die Papiererkennung bei Speicher &9F12 mit 57 gepoked (siehe Listing 1).

Eike Biel
Hamburg

Ewiger Kalender

Ihre Antwort in Heft 4/86, auf die Anfrage zum Kalenderprogramm, ist zwar grundsätzlich richtig, kann aber so dem von Herrn Funke geschilderten Problem nicht beikommen. Vorausgesetzt, daß die Zeilennummern unverändert sind, weist die Fehlermeldung "Improper argument in 3030" nämlich auf den Befehl "SYMBOL AFTER 91" hin. Diese Anweisung kann aber nur dann den besagten Fehler verursachen, wenn irgendwann vorher die Speichergrenze durch eine nicht geschlossene Datei oder durch den MEMORY-Befehl verschoben wurde (siehe dazu "Der gläserne CPC" in Heft 12/85). Diese Verlegung kann auch durch einen Probelauf des Kalenderprogramms hervorgerufen worden sein, da in der Initialisierungsroutine eine entsprechende Befehlsfolge existiert (Zeile 3090). Herr Funke sollte deshalb versuchen, das Programm direkt nach dem Einschalten zu starten; in diesem Fall sollte der Fehler nicht mehr auftreten.

Hierzu noch ein paar Tips: Es ist nicht nötig, das Programm immer neu zu starten, wenn man sich durch (E) zu früh verabschiedet hat. Es kann ganz einfach durch "CONT" wieder zum Leben erweckt werden, da ja alle wichtigen Werte erhalten bleiben und der CPC danach die Sprunganweisung "ON wahl GOTO" abarbeitet (wer ganz von vorn anfangen will, kann dies auch mit "RUN 3110" tun, womit auch die o.g. Fehlermeldung vermieden wird). Die REM-Zeilen kann man

Listing 1

```
   1 REM  "NLQ aus"   -> löschen
   2 POKE &9F2F,00: POKE &9F30,00: POKE
     &9F31,00
   3 REM  ELITE ausschalten (PICA ein)
   4 POKE &9F1B,80: POKE &9F1C,00
   5 DELETE 1 - 5
  80 casin=&9E02: cata=&9E3E: cset=&9F15:
     cabs=&9F19: elite=&9F1B
1340 POKE elite,77: gosub 1830: goto 1290
1350 POKE elite,80: gosub 1830: goto 1290
1830 PEN 2: LOCATE 16,4: PRINT "ELITE"
1835 LOCATE 22,4: IF PEEK(elite)=77 THEN
     PRINT "ein" ELSE PRINT "aus"
1840 LOCATE 16,6: PRINT "PICA"
1845 LOCATE 22,6: IF PEEK(elite)=80 THEN
     PRINT "ein" ELSE PRINT "aus": RETURN
```

Wir setzen Maßstäbe: 1 Megabyte = 758 DM



Für alle CPC's (464 + DDI-1, 664, 6128)

vortex Floppy Disk System 1-X.

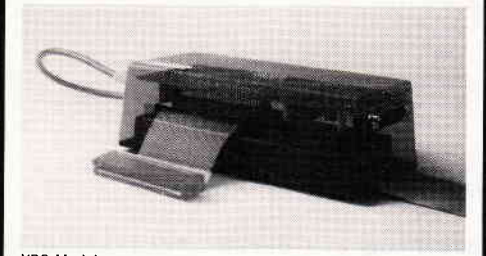
(F1-X (RS) = 5 ¼" und M1-X (RS) = 3 ½")

Damit präsentieren wir eine professionelle Systemlösung für alle CPS's, welche schon ein 3" Laufwerk haben, aber den Standard brauchen (5 ¼" bzw. 3 ½").

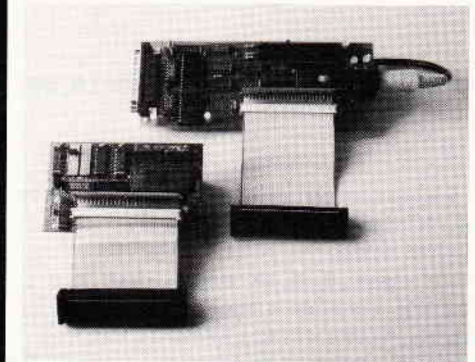
- Die F1-X (M1-X) ist kein Zweitlaufwerk, sondern Ihr **Systemlaufwerk.**
- Per Kommando "IX" vertauschen Sie nach Belieben die Reihenfolge von 3" und F1-X bzw. M1-X.
- Der Befehl "ICPM,1" oder "ICPM,2" ermöglicht Ihnen erstmals CPM aus beiden Laufwerken gleichberechtigt zu starten.
- Ihr 3" Laufwerk und die F1-X (M1-X) unterscheiden sich zwar wesentlich in Datendichte und Datenformat (180 Kb zu 708 Kb), das Kopieren zwischen beiden geschieht jedoch so reibungslos und glatt, als gäbe es diesen Unterschied nicht.
- Sie wählen per Kommando zwischen **"Amsdos"** und **"VDOS 2.0"** als Betriebssystem – Sie verfügen über beides nach Ihrer Wahl.
- Als herausragende Leistung bietet Ihnen **VDOS 2.0**:
  - 128 Directory Einträge
  - direkter relativer Dateizugriff unter Basic. 16 Dateien können gleichzeitig geöffnet sein.
  - ROM residenter Monitor + Diskeditor
  - mit "I Format" formatieren Sie direkt unter Basic
  - mit "I Code,«var»" realisieren Sie einen Programmschutz mit persönlichem Passwort
  - direkte Parametereingabe bei RSX-Befehlen Beispiel: **I ERA,"Test.Bas"** statt wie bislang gewohnt: **"A$="Test.Bas", I ERA, eA$**
- **RS 232** wird benötigt? Dann wählen Sie die **F1-XRS** oder die **M1-XRS.** Ein kompaktes RS 232 Modul ist bei diesem Typ integriert. Sie können aber auch Ihre F1-X nachträglich damit ausrüsten.
- Übrigens: Das **vortex RS-Modul** ist voll **Amstrad Port-kompatibel.** Die im Amsdos vorhandene RS-Treibersoftware arbeitet sofort.
- Einfachste Montage: X- bzw. XRS-Modul und 3" Controler werden einfach zusammengesteckt.

vortex F1-X (RS) und M1-X RS)

Das Besondere unserer neuen X-Lösung ist das sogenannte **X(RS)-Modul** (siehe Abbildung). Es handelt sich dabei um unseren Diskcontroler, der um all die Komponenten abgemagert wurde, welche im Schneider Controler schon gegeben sind.



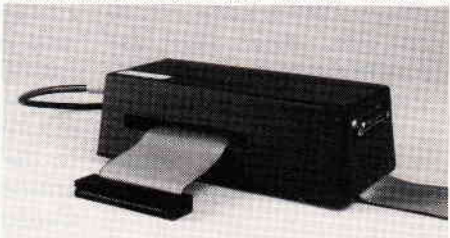
XRS-Modul ▲
X-Modul und XRS-Modul ohne Gehäuse ▼



Vortex X(RS)-Modul + Schneider Controler ergibt die neue Qualität! Ihr besonderer Knackpunkt: Ihr **Amsdos + Logo** im 3" Controler bleiben Ihnen vollgültig erhalten, unser Betriebssystem mit seinen erweiterten Features gewinnen Sie dazu und damit die Fähigkeit, ein **Megadrive** zu verwalten. Bei dem Design des **X-Konzepts** wurde schon alles vorgesehen zum Betrieb einer Standard-**RS 232 Schnittstelle.** Zum Beispiel muß nicht, wie sonst üblich, ein weiteres Netzteil zugeschaltet werden – das im Laufwerksgehäuse integrierte Netzteil übernimmt die Versorgung. Die Schnittstelle selbst befindet sich im X-Modul, an dessen Schmalseite der RS-Konnektor austritt. Eine an Kompaktheit und Eleganz – so behaupten wir – kaum zu überbietende Lösung: Sie haben ein kombiniertes RS-Controler Modul. Eine Einheit, mechanisch kompakt und stabil.



6128 und F1-X Floppy ▲
Trickfoto des XRS-Moduls und des 3" Schneider Controlers ▼



vortex X (RS)-Modul + Schneider Controler

Soll die Schnittstelle erst später nachgerüstet werden, so tauschen wir Ihr X-Modul gegen ein XRS-Modul zum Differenzpreis plus DM 10,– Bearbeitungsaufwand um!
Für Leute, denen noch 16 K ROM fehlen:
Im X-Modul wird ein 256 K ROM verwaltet, 128 K sind hier prinzipiell noch frei. ROM Adresse kann frei über Jumper gewählt werden.

Mit unserem **1-X (RS)-Konzept** bieten wir ein Preis-Leistungsverhältnis bei welchem wir mit Recht sagen dürfen:

Vergleichen Sie und sagen Sie uns: Wer kann mehr?

M1-X (3 ½") 758,–*
F1-X (5 ¼") 758,–*
M1-XRS (3 ½") 858,–*
F1-XRS (5 ¼") 858,–*
(XRS-Modul: auf Anfrage)

* unverbindliche Preisempfehlung

Achtung: Bitte geben Sie bei Kauf oder Bestellung unbedingt Ihren Computertyp an!

Für soft- und hardwaretechnische Fragen im Zusammenhang mit unseren Produkten haben wir eine **User-Sprechstunde** eingerichtet: Montag + Donnerstag von 18 – 21 Uhr stehen wir Ihnen telefonisch zur Verfügung! Telefon (0 71 39) 21 60 und 79 60 sowie (07 11) 777 55 76

vortex Computersysteme · Vertriebs GmbH
Klingenberg 13 · 7106 Neuenstadt 5
Telefon (0 71 39) 21 60 und 79 60 · Telex 7 28 915 trond





ohne Probleme weglassen, wenn man auf die folgende Programmzeile achtet:
*1620 POKE 383,&1E:znr= 2720+(x MOD 28)*10*
Hier erfolgt nämlich die Berechnung der Zeilennummer für den relativen RESTORE-Befehl, und zwar von der ersten DATA-Zeile der Startwertetabelle ausgehend. Deshalb muß man nach der RENUMerierung nur die neue Zeilennummer dieser DATA-Zeile einsetzen, und schon läuft das Programm auch ohne REM-Zeilen.

Matthias Iwachiw
Nieste

Context I

Ich habe mich, wie viele andere Leser wahrscheinlich auch, sehr über das Textprogramm CONTEXT gefreut. Mich begeistert vor allem die Schnelligkeit und Vielseitigkeit des Programms.
Da ich den Drucker STAR SG-10 besitze, mußte ich eine Druckeranpassung vornehmen. Die beigelegte Liste enthält alle Programmänderungen, die ich all den Lesern zur Verfügung stellen möchte, die mit diesem Drucker arbeiten (siehe Anpassung zu Context):
Wenn man anhand dieser Liste alle Änderungen durchgeführt hat, läuft der Druckvorgang genauso ab, wie in der Programmbeschreibung. Was mir leider nicht gelungen ist, daß verschiedene Schrifttypen bei NLQ ausgedruckt werden, dies ist nur beim Entwurfsdruck möglich. Vielleicht gibt es aber auch einen Leser, der dieses Problem noch lösen kann!

Markus Neugebauer
Reutlingen

```
            Anpassung von CONTEXT an den Drucker
                        STAR   SG-10

1. Programm CONTEXT Laden durch RUN ""

2. Sobald das Menü erscheint, das Programm durch  Programmpunkt BASIC
   verlassen

3. Erstellen der Zeile 115:

   115 POKE &9F1B,66:POKE &9F30,66:POKE &9F12,8:POKE &9F15,55

4. Ändern der folgenden Zeilen:

     80 casin=&9E02:cata=&9E3E: cset=&9F16 :cabs=&9F19:nlq=&9F1C
   1340 POKE nlq,4 :GOSUB 1830:GOTO 1290
   1350 POKE nlq,5 :GOSUB 1830:GOTO 1290
   1360 IF PEEK (cset)=0 THEN POKE cset,2 ELSE POKE cset,0
   1390 POKE mzei,n:POKE cabs,1728/n :GOSUB 1850:GOTO 1290
   1830 PEN 15-PEEK(nlq) :LOCATE 16,4:PRINT"Briefqualität"
   1840 PEN 6+PEEK(nlq) :LOCATE 16,6:PRINT"Entwurfsdruck":RETURN
   1890 IF PEEK(cset)= 0 THEN PRINT"I" ELSE PRINT "D"
   1920 IF PEEK(cset)= 0 THEN 2050

5. Programm durch RUN starten und Funktionen überprüfen

6. Programm dann wieder über BASIC verlassen

7. Programm abspeichern durch:

   SAVE "CONTEXT":SAVE "CONTEXT.BIN",b,&9300,&D00

Bemerkung:  Verschiedene Schrifttypen können nur beim Entwurfsdruck
            realisiert werden, nicht aber bei NLQ!
```

Joy-Funktion

Ich möchte hiermit auf eine sehr angenehme Eigenschaft des Schneider-Locomotiv-Basics hinweisen.
Der selten benutzte Befehl "DEF FN" eignet sich nicht nur zur Definition von mathematischen Funktionen, sondern es können auch Basic-Befehle oder ganze Befehlsfolgen (durch Doppelpunkt getrennt) definiert werden. Dies ist z.B. in solchen Programmen, die wahlweise mit Joystick oder den Cursortasten arbeiten, sehr nützlich. Hier kann dann am Anfang des Programms direkt nach der Abfrage eine User-Funktion für die unterschiedlichen Inkey-Nummern definiert werden. Zum Beispiel wie in Listing 2.

Listing 2

```
10 'Anfang des Programms
20 PRINT"<C>ursortasten oder <J>oystick?"
30 a$=INKEY$:a$=UPPER$(a$):IF a$<>"J" OR a$<>"C" THEN 30 ELSE 40
40 IF a$="J" THEN DEF FNauf=INKEY(72):DEF FNab=INKEY(73):DEF FNlinks=
   INKEY(74):DEF FNrechts=INKEY(75) ELSE DEF FNauf=INKEY(0):DEF FNab=
   INKEY(2):DEF FNlinks=INKEY(8):DEF FNrechts=INKEY(1)
50 'Hier schließt das Hauptprogramm an
```

Statt umständlicher Abfrage von Flags und dadurch bedingte Unterprogrammsprünge, kann nun der Joystick- bzw. Cursortastenstatus einfach über die User-Funktion FNauf, FNab, etc. abgefragt werden. Wer will, kann auch noch eine Abfrage des Feuerknopfes bzw. der COPY-Taste hinzufügen.

Wolfram Arnold
Usingen

MAIL232.COM

Seit kurzer Zeit besitze ich den PCW 8256 von Schneider. Ich habe auch den DFÜ-Artikel gelesen. Daher habe ich versucht, das MAIL232.COM, das auf der LocoScript-Diskette versteckt ist, herauszuholen.
Die Datei MAIL232.COM ist eine SYS-Datei. Ich bediente mich dem SET-Dienstprogramm.
Mein Vorgehen:
Ich habe eine Backup von LocoScript gemacht und einige Programme gelöscht (Platzmangel).
Mit dem PIP-Kommando habe ich die Dienstprogramme SET.COM und DIR. COM in die RAM-Disk geladen.
Wenn man dann "A >m:DIR eingibt, kommt im Directory

die Datei mit MAIL232.COM 4K 32 SYS RW. *Mit SET. COM habe ich diese Datei in ein* MAIL232.COM 4K 32 DIR RW *umgewandelt.*
Eingabe: A> m.SET mail 232.COM ÄDIRÜ. *Somit konnte ich dann die Datei auf jede andere Diskette kopieren, und mit PROFILE.SUB starten lassen.*
Ich hoffe, mit dieser Information gedient zu haben.

Graman Engelbert, Linz

Quickplan mit X- & Y-Koordinaten

Wenn man von einer mit Quickplan erstellten Schaltung, die über mehrere Blätter geht, eine Hardcopy machen will, ist es einfacher, wenn die Übergänge von Blatt zu Blatt gleich sind. Dazu dient folgende Programmzeile:
WINDOW # 3,16,26,1,1: PRINT 3,X;Y.
Diese beiden Befehle erhalten die Zeilennummern 1695, 1795, 1975. Dadurch wird im Zeichnen-, Lauf- und Radiermodus immer der aktuelle X/Y-Stand des Cursors angezeigt. Das ermöglicht u.a. auch sehr einfaches Zeichnen von IC's.

Armin Fendl, Bergenhausen

Stellungnahme

Bezugnehmend auf den Leserbrief des Herrn Ebert/ Duisburg, erschienen in Ausgabe 4/86, möchten wir als Hersteller folgende Stellung nehmen:
Es ist bzw. war richtig, daß unser erweitertes Betriebssystem nicht mit dem Patch der Fa. Vortex zusammenarbeitete, da dieser Patch keine Ergänzung des Betriebssystemes bewirkte, sondern ein komplett neues BIOS, sprich Betriebssystem, darstellt.
Da wir jedoch auch unseren Kunden den Weg in das neue CP/M 62K von Vortex ebnen wollten, liefen die zur Anpassung unserer Erweiterung nötigen Arbeiten bereits seit Dezember letzten Jahres auf Hochtouren. Die Arbeiten konnten Anfang März dieses Jahres abgeschlossen werden. Nunmehr steht dem Betrieb unserer Controller Systeme sowie den Cumana Up-Dates unter Vortex CP/M 62K nichts mehr im Wege. Benutzer dieser Systeme, die bisher noch keinen Nachtrag zu Ihrer Systemdiskette erhalten haben, können diese Anpassung direkt über uns bzw. ihren Händler vor Ort kostenlos beziehen.

Data-Service Bonn
Bornheimer Str. 82
5300 Bonn 1

Zweitlaufwerk

1. *Im Sonderheft 2 sind Artikel über den Anschluß von Zweitlaufwerken enthalten. Leider kamen diese für mich zu spät, da ich bereits ein Hitachi-Laufwerk an meinen CPC 6128 mit einigen Mühen angeschlossen hatte. Ihre Angaben hierzu, z.B. auf Seite 83, sind für den 6128 nicht korrekt. Der Belegungsplan im Schneider-Handbuch ist ebenfalls falsch!*
Richtig ist folgendes: Am Centronics-Stecker am 6128 sind die Anschlüsse 1 und 19 nicht belegt! Die Anschlüsse 2 – 18 sind Erde, die Anschlüsse 20 – 36 sind mit den Anschlüssen 2 – 34 (wie von Ihnen beschrieben) des Zweitlaufwerkes zu verbinden. Die Zeichnung auf Seite 84 ist entsprechend zu korrigieren. Das Ready-Signal der Diskettenstation liegt an Pin 36 und der Motorschalter an Pin 27 des Centronics-Steckers. Da die Anschlüsse am Centronics-Stecker über Kreuz liegen, muß hier gelötet werden. Für den Floppyanschluß kann ein Quetschstecker verwendet werden.
Daß nach den Angaben der Artikel eine Zweitstation nicht in Gang gebracht werden kann, liegt offensichtlich daran, daß der 6128 im Gegensatz zum 664 einen Centronics-Stecker hat.

2. Zum RSX-Programm *"Restore" in Heft 4/86, Seite 106, möchte ich bemerken, daß es so nicht funktioniert. Zeile 280 "Memory &A9FF" muß zur Fehlermeldung "Memory full" führen. Aber auch eine Herabsetzung auf &9fff führt nicht zum Erfolg: Die Meldung lautet "Data exhausted in 350". Gibt man in Zeile 350 den normalen Befehl "Restore 180" ein, so läuft das Programm ohne Fehler.*

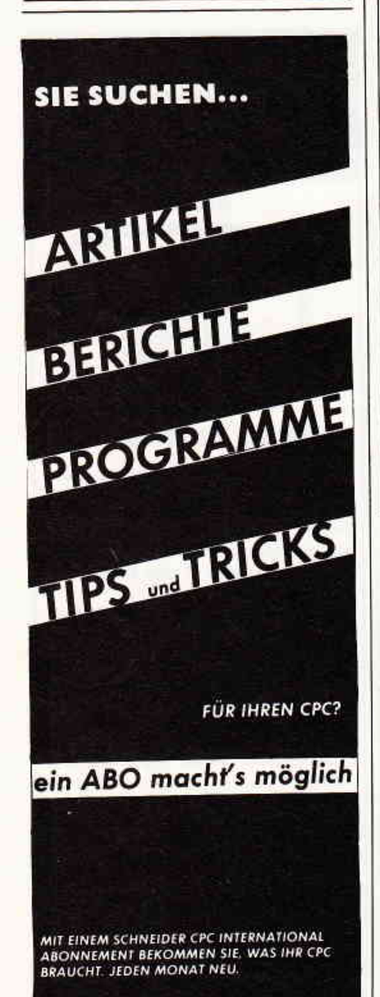
Dr. Dieter Bauer, Krefeld-Bockum

CP/M und seine Möglichkeiten

Im zweiten Teil Ihrer Serie über CP/M und seine Möglichkeiten befassen Sie sich mit dem Gespann MBASIC und BASCOM. Sie schreiben, daß Schneider-Sound und -Grafik leider nicht zur Verfügung stehen. Gleiches gilt leider auch für den Microsoft FORTRAN-Compiler F80. Da FORTRAN immer noch sehr vital ist und anerkannt schnellen Code erzeugt, hatte ich schnell den Wunsch, diesen Mangel abzustellen. Für den "464" gestaltete sich das, aufgrund der glatten Firmwareschnittstellen und der Ausbaufreundlichkeit der Compiler, leicht.
Ich selbst war zwar nur an der MODE-2-Grafik interessiert, weitere Anbindungen sind aber sicher möglich. Eigentlich müssen lediglich die passenden Register mit Parametern versorgt und die richtigen Routinen angesprungen werden.
Das beiliegende Programm für den M80 übernimmt diese Aufgabe. Das erzeugte REL-File kann entweder für sich bleiben oder in eine bestehende Bibliothek eingefügt werden. Es kann mit BASCOM oder F80 gleichermaßen zusammenarbeiten. Dramatisch ist der Gewinn an Ausführungsgeschwindigkeit allerdings nicht.

TITLE CPCGRAF
Dieses kurze Programm macht einige wichtige Schneider-Grafikroutinen den Mirosoft Compilern F80 und BASCOM unter CP/M zugänglich.





Die mit dem Macro-80 erzeugte Datei CPCGRAF.REL kann entweder mit L80 eingebunden oder mit LIB80 in die Standardbibliothek eingefügt werden. Der Aufruf der Routinen läuft dann in FORTRAN und BASIC so, wobei alle Argumente als Integer zu deklarieren sind:

CALL GRINIT
Initialisieren der Grafik und Vorbereiten für Zugriff auf die Firmware. Muß erster Grafik-Call sein!

CALL CLRSCR
Bildschirm löschen

CALL MOVE (X,Y)
Wie im Schneider BASIC

CALL PLOT (X,Y)
s.o.

CALL DRAW (X,Y)
s.o.

CALL DRAWR (X,Y)
s.o.

CALL PEN (P)
s.o.

```
GETXY   MACRO                 Koordinaten holen
        PUSH DE               Pointer auf Y retten
        PUSH HL               Pointer auf X retten
        POP IX
        LD  E,(IX+0)
        LD  D,(IX+1)          X-Koordinate in DE
        POP IX
        LD  L,(IX+0)
        LD  H,(IX+1)          Y-Koordinate in HL
        ENDM

MOVE:   GEXY                  Aufruf CALL MOVE (X,Y)
        CALL GRAMOV
        RET

PLOT:   GETXY                 Aufruf CALL PLOT (X,Y)
        CALL GRAPLO
        RET

DRAW:   GETXY                 Aufruf CALL DRAW (X,Y)
        CALL GRADRA           'CPC DRAW ABSOLUT'
        RET

DRAWR:  GETXY                 Aufruf CALL DRAW (X,Y)
        CALL GRADRR           'CPC DRAW RELATIVE'
        RET

PEN:    LD A,(HL)             Aufruf CALL PEN (P)
        CALL GRAPEN
        RET
        END
```

```
Z80
ENTRY GRINIT, CLRSCR, MOVE, PLOT, DRAW,
DRAWR, PEN
MEMKON   EQU   07F86H   Speicherkonfiguration
GRAINI   EQU   0BBBAH   Grafik initialisieren
GRACLR   EQU   0BBDBH   Grafikschirm löschen
TXTCLR   EQU   0BB6CH   Textschirm löschen
GRAMOV   EQU   0BBC0H   Koordinate verschieben
GRAPLO   EQU   0BBEAH   Punkt absolut plotten
GRADRA   EQU   0BBF6H   Linie absolut ziehen
GRADRR   EQU   0BBF9H   Linie relativ ziehen
GRAPEN   EQU   0BBDEH   Schreibfarbe wählen

GRINIT:  EXX            Speicherkonfiguration
                        nach BC'
         LD BC,MEMKON
         EXX
         CALL GRAINI    Grafik initialisieren
         CALL GRACLR
         CALL TXTCLR    Bildschirm gelöscht
         RET

CLRSCR:  CALL GRACLR
         CALL TXTCLR    Bildschirm gelöscht
         RET
```

Die Microsoft-Compiler übergeben nicht die Argumente selbst, sondern Pointer auf die betreffenden Speicherzellen. Hat man weniger als 4, so werden alle Pointer in den CPU-Registern übergeben. Vor dem Aufruf der Firmware müssen die wirklichen Argumente geholt und in die passenden Register geschrieben werden.

PROGRAMM HUT
Dieses FORTRAN Programm zeigt die Möglichkeiten der Erweiterung. Es zeichnet eine bekannte Funktion von zwei Variablen, die an einen Sombrero erinnert.

INTEGER PY,PX,PT,T(600)

T(600) ist ein Kontrollfeld für unsichtbare Punkte. Die übergebenen Parameter sind implizit deklariert.

```
      DO 1 PT=1,600
        T(PT)=-200
1     CONTINUE

      CALL GRINIT
      F=150.0
      DO 2 IY=0,100
        Y=-10.0+IY*0.2
        DO 3 IX=0,40
          X=-10.0+IX*0.5
          R=SQRT(X*X+Y*Y)
          Z=F*SIN(R)/R
          PX=INT(20*(X+Y/2))
          PY=INT(10*Y+Z)
          PT=PX+300
          IF (PY .LT. T(PT)) GO TO 3
          IPX=PX+300
          IPY=PY+200
          CALL PLOT (IPX,IPY)
          T(PT)=PY
3       CONTINUE
2     CONTINUE
      END
```

Joachim Sieben,
Großen-Buseck

SP-Hardcopy

Nachdem ich mir den Drucker Seikosha SP-1000 gekauft hatte und die erste Hardcopy erstellen wollte, funktionierte soviel wie bei Ihnen im Test – fast nichts – nur das linke Drittel des Bildes und sonst irgendwelche Zeichen.
Nach halbtägiger Suche am und im Drucker (dabei reparierte ich nebenher den defekten Schalter für die Zeilenvorschubeinstellung – Garantie ade) entdeckte ich den Fehler, und zwar im Handbuch. Die ganze Software des Druckers ist perfekt auf den CPC ausgelegt, bloß das Handbuch hat einen kleinen Hänger: Im ganzen Drucker gibt's kein achtes Bit. Für die Berechnung der Übergabeparameter braucht man's aber!?

*Berechnet man jetzt die Übergabeparameter – abweichend vom Handbuch – nach Punktanzahl=n1 + 128*n2, dann stimmt alles.*
Mit diesem Drucker kann also jede beliebige Punktanzahl gedruckt werden, auch 640 für den CPC. Die Änderungen in den Hardcopy-Routinen sind auch einfach: Irgendwo in der Routine, meistens ziemlich am Schluß, befindet sich die Byte-Übergabefolge 1B 4B 02 7F. Ändert man das 02 in 04, dann arbeiten die Routinen einwandfrei mit dem Seikosha.

Dipl.-Ing. Frank Oesterle, Laichingen

Ewiger Kalender II

Leider ist mir, wie die Anfrage von Herrn Freisleben zeigt, bei der Erstellung des Kalenderprogramms ein Fehler unterlaufen, den ich in den Testläufen übersehen habe. Dieser Fehler hängt natürlich mit dem relativen RESTORE-Befehl zusammen, aber auch mit ungenügenden Informationen über den Leerzeichen-POKE &AC00,255, den ich beim Eintippen verwendet habe. Ich mußte, nachdem ich das Programm jetzt noch einmal durchsah, feststellen, daß dieser Befehl nicht nur die sichtbar überflüssigen Leerzeichen einer Eingabezeile entfernt, sondern auch jene, die ein Befehlswort wie RESTORE von der nachfolgenden Zeilennummer trennen. Dieser Effekt ist auf dem Bildschirm leider nicht zu erkennen, da der BASIC-Interpreter die Zeile beim LISTen wieder mit den trennenden Leerzeichen versieht. Man kann ihn nur feststellen, wenn man sich die Zeilen entweder mit einer PEEK-Schleife aus dem Speicher holt oder sich deren Checksummen ausgeben läßt.. Das ist auch der Grund, weswegen das Programm in der Listing-Version nicht laufen kann: Da die RESTORE-Routine infolge des Leerzeichen-POKEs auch beim Probelauf Zeilensalat produzierte, habe ich die Adressen der dazugehörenden POKE-Befehle verändert. Es lief dann reibungslos ab, aber durch die Korrektur bildete sich ein "Zwittertext": Zeile 1000 der Cassettenversion beinhaltet keine Leerzeichen, Zeile 1010 (die Error-Zeile) bekam jedoch das Leerzeichen zwischen RESTORE und der Zeilennummer zurück. Dieser Unterschied ist im Listing logischerweise nicht zu sehen und kann deshalb auch nicht eingegeben werden.
Um das Programm funktionsfähig zu machen, sind aber nur ein paar Änderungen nötig: Geben Sie die Zeilen 1000 und 1010 neu ein, und zwar so, wie sie im Listing stehen. Ändern Sie dann folgende Zeilen:

```
1620 POKE 384,&1E:znr=
     2720+(x MOD 28)*10
1630 POKE 385,znr MOD
     256:POKE 386,INT
     (znr/256)
```

Sie müssen also nur die POKE-Adressen um 1 erhöhen, um den Fehler zu beseitigen. Verwenden Sie aber nicht den Leerzeichen-POKE, da ich in diesem Fall für nichts garantieren kann.

Matthias Iwachiw, Nieste

Zahl in String?

Ich habe seit Weihnachten '85 einen Schneider CPC 464. Da das sonst gute Handbuch auch Fragen offen läßt, wende ich mich eben an Sie.
Ich möchte eine Zahl in einen String bekommen. Also: Ich habe x=5 und möchte, daß der String a$ eine 5 enthält, die 5 aus der Variablen x. Ich kann ja diese zwei Dinge nicht anwenden: a$=x, oder a$="x". Das erste ergibt einen Fehler, und beim zweiten habe ich in a$ nur ein "x" stehen. Wie mache ich es nun, daß a$ eine "5" enthält?

Christof Pflumm,

CPC:
Der Befehl STR$ wandelt eine Zahl in einen String. Haben Sie also: x=5, so steht nach Eingabe von: A$= STR$(x) in der Variable A$ der Wert "5".

Tabelle Sequenzen

```
&9F11  DB 27,64,0    -  Normierung des Druckers /[1]
&9F14  DB 27,82,n    -  Zeichensatz (n=0 oder n=2)
&9F17  DB 27,51,n    -  n/216" Zeilenabstand
&9F1A  DB 27,x,n     -  Kursiv Schrift ein/aus (x=52 oder x=53)/[2]
&9F1D  DB 27,45,0    -  Unterstreichen aus
&9F20  DB 27,87,0    -  vergrößerte Schrift
&9F23  DB 18,0,0     -  Zeichenverdichtung aus
&9F26  DB 27,84,0    -  hoch-/tiefgestellte Indizes aus
&9F29  DB 27,70,0    -  Zeichenhervorhebung aus
&9F2C  DB 27,72,0    -  Doppelanschlag aus

&9F2F  DB 27,53,0    -  Kursiv Schrift aus
&9F32  DB 27,45,1    -  Unterstreichen ein
&9F35  DB 27,87,1    -  vergrößerte Schrift ein
&9F38  DB 15,0,0     -  Zeichenverdichtung ein
&9F3B  DB 27,83,0    -  hochgestellte Indizes ein
&9F3E  DB 27,83,1    -  tiefgstellte Indizes ein
&9F41  DB 27,69,0    -  Zeichenhervorhebung ein
&9F44  DB 27,71,1    -  Doppelanschlag ein
&9F47  DB 13,10,255  -  Ende der Zeile
```

Tabelle Basicteil

```
  80  cset=&9F16
1340 POKE nlq-1,52:POKE nlq,1:GOSUB 1830:GOTO 1290
1350 POKE nlq-1,53:POKE nlq,0:GOSUB 1830:GOTO 1290
1360 IF PEEK(cset)=2 THEN POKE cset,0 ELSE POKE cset,2
1890 IF PEEK(cset)=0 THEN PRINT"I" ELSE PRINT"D"
1920 IF PEEK(cset)=2 THEN 2050
```

Context

Ich habe das Programm "Context" aus Heft Nr. 4/86 abgetippt und habe es auf dem Drucker von EPSON FX 80+ angepaßt.

Ich möchte Ihnen mit diesem Schreiben die Sequenzen für diesen Drucker mitteilen.

Es sind folgende Sequenzen nötig (s. Tabelle Sequenzen).

1. *Die Normierung muß durchgeführt werden, da sonst der Drucker beim Drucken den automatischen Seitenvorschub vollzieht.*
2. *In Adresse &9F1C wurde das n=0 und n=1 beibehalten, um im Basicteil die gleichen Farbenwerte zu erhalten.*

Weiterhin wurden im Basicteil die folgenden Zeilen geändert (s. Tabelle Basicteil).

Ich möchte noch hinzufügen, daß ich mit diesem Programm sehr zufrieden bin.

Reinhold Stollarski, Frankfurt

Kennworteingabe

Ich möchte die Benutzung eines Programmes nur nach Eingabe eines korrekten Kennwortes gestatten. Dabei soll die Eingabe auf dem Bildschirm nicht sichtbar sein. Soweit ich es verstanden habe, soll dies mit Hilfe der INKEY$ in Basic möglich sein – aber wie? Ein Beispielprogramm würde mir hier sehr weiterhelfen.

Ludwig Schwab, Lünen

CPC:
Ein kleines Beispiel, welches das Prinzip einer Eingabe ohne Bildschirm-Echo zeigt:

```
10 'Bsp. Kennworteingabe ohne Bildschirm-Echo
20 'Das richtige Kennwort kann als Zeichen-
30 'kettenkonstante im Programm definiert
40 'sein oder über entspr. Algorithmus
50 'generiert werden. Im zweiten Fall muß
60 'das vom Anwender eingegeben Kennwort
70 'ebenfalls umkodiert werden.
80 'Zusätzlich könnte man die Fehleingaben
90 'zählen und nach überschreitung das
100 'Programm abbrechen oder ähnliches...
110 '
120 kw$="Test" 'oder Erzeugung durch Algor.
130 einkw$="" 'nimmt Eingabe auf.
140 PRINT:PRINT"Kennwort ? ";
150 'auf Eingabe eines Zeichens warten:
160 ch$=INKEY$:IF ch$="" GOTO 160
170 IF ch$=CHR$(13) GOTO 220 'Eingabe beendet
180 einkw$=einkw$+ch$ 'Kennwort aufbauen
190 PRINT"."; 'nur zur Orientierung
200 GOTO 160 'nächstes Zeichen
210 '
220 'hier eingegebenes Kennwort notfalls
230 'entspr. umkodieren.
240 '
250 'bei falschem Kennwort wiederholen:
260 IF einkw$<>kw$ GOTO 130
270 '
280 'Bei richtigem Kennwort folgendes
290 'Programm ausführen
```

```
Context  Listing
4440 DATA 5148
4450 DATA 39,ed,38,ee,31,ef,1e,1d
4460 DATA 1f,1e,27,1f,11,0c,13,12
4470 DATA 00,00,00,00,00,00,00,00
4480 DATA 00,00,00,00,00,1b,52,00  Unterstreichen aus
4490 DATA 00,00,00,00,00,00,00,00
4500 DATA 00,1b,26,00,1b,26,00,00  Fett-/Doppeldruck aus
4510 DATA 00,00,1b,45,00,00,00,00  Unterstreichen ein
4520 DATA 00,00,00,1b,44,00,1b,55  Index hoch/tief ein
4530 DATA 00,1b,57,00,1b,46,00,0d  Fett-/Doppeldruck ein
4540 DATA 0a,ff,04,08,01,10,20,20  Belegung der Dreiecke
4550 usw. unverändert
```

Context

Ihr Programm "Context" aus Heft 4/86 habe ich abgetippt und es leistet mir seither gute Dienste. Mein Matrixdrucker (Brother 1009) führt auf Anhieb alle ihm möglichen Druckbefehle klaglos aus.
Für meinen Typenraddrukker (Brother HR 10) paßte die Installation allerdings überhaupt nicht. Als blutiger Anfänger in Sachen Programmieren bzw. Programmänderungen war ich zunächst ratlos. Dann fiel mir auf, daß in Block 25 des Maschinenprogramms die Druckbefehle liegen. Ich habe nun Block 25 folgendermaßen abgeändert (laut Druckerhandbuch) (s. Listing zu Context).
Auf Dreieck nach links liegt Doppelanschlag.
Auf Dreieck nach rechts liegt Fettdruck (Zeichenhervorhebung).
Auf Dreieck nach oben liegt Hochstellung.
Auf Dreieck nach unten liegt Tiefstellung (jeweils eine halbe Zeile). Unterstreichen funktioniert wie bei der Standardeinstellung (allerdings nur unter den Zeichen, Leerstellen werden nicht unterstrichen). Bei den hoch- und tiefgestellten Indizes bin ich aber mit meiner Weisheit am Ende. Da der HR 10 keinen auffindbaren Befehl zum Rückschalten der halben Zeilenschaltung hat, gibt es Schwierigkeiten. Der erste Befehl wird befolgt (mit Rückstellung durch das Gegenzeichen). Will ich aber nach einer Hochstellung einige Zeichen weiter eine Tiefstellung anbringen, druckt der HR 10 eigensinnig trotzdem alles hoch. Daß es doch richtig geht, beweist er beim WordStar (Einstellung Diablo 630). Irgendwo habe ich bei der Installation einen kapitalen Fehler sitzen.
Da beim Typenraddrucker die hochgestellte 2 und 3 vorhanden sind, ist der Schaden nicht allzu groß, denn nur tiefgestellte Indizes verkraftet meine Einstellung ja.
Trotzdem wäre ich für einen Ratschlag sehr dankbar.

Klaus Sippel, Oberstdorf

Hitachi-3"-Laufwerk an Joyce?

Eine Frage, die vielleicht mehrere Joyce-Besitzer betrifft. Von der Firma Völkner habe ich mir ein Hitachi-3"-Laufwerk gekauft. Leider fehlt es mir an Erfahrung, wie dieses Laufwerk an den Joyce angeschlossen werden kann bzw. als Zweitlaufwerk eingebaut werden muß. Ich würde mich freuen, wenn technisch versiertere Leser mir bei diesem Problem helfen könnten.

Karl Zwickl, Postfach 1325, 8263 Burghausen

Utopia

Vor knapp einem Jahr habe ich Ihr Magazin erstmals gekauft. Das Heft machte einen amateurhaften Eindruck, z.B. verglichen mit dem englischen "Amstrad User". In der Zwischenzeit habt Ihr gewaltige Fortschritte gemacht und den (akzeptablen) Preisaufschlag werde ich kompensieren, indem ich ein anderes Schneider-Magazin nicht mehr jeden Monat kaufe.
Gefreut habe ich mich über Ihre Besprechung von "Utopia" im Mai-Heft. Aber auch gewundert über die Schlußbemerkungen:
1. Sie schreiben, das zweite ROM-Modul, "Protext", sei bereits angekündigt. Dieses Textverarbeitungsprogramm ist in England bereits seit September 1985 erhältlich. Ich benutze die ROM-Version nun schon sieben Monate und bin begeistert davon. Vor allem die Geschwindigkeit der Operationen ist fantastisch.
2. Seit April bietet Arnor zwei Ergänzungsprogramme an: a)"PROSPELL", ein englisches Rechtschreibungs-Prüfprogramm
b)"PROMERGE", ein Dateimischprogramm für Serienbriefe etc., welches in der ROM-Version zusätzliche Erweiterungen zu "Protext" enthält: Hintergrund-Druck, gleichzeitiges Bearbeiten von zwei Dateien, Bearbeiten von mehreren Spalten, Schreibmaschinen-Modus, Rechenfunktion etc. Auch werden die zusätzlichen 64K Speicher beim 6128 genutzt, dadurch können Dateien bis zu 80K im Speicher gehalten werden.
3. Eine kleine Korrektur: Die Arnor-Epromkarte, die in den Slot der Platine gesteckt werden kann, enthält nur vier, nicht fünf, leere Sockel.
4. Die neue Version der Zusatzplatine (AD2) paßt auch an den 6128. Sie ist mit Erstbaustücken von MAXAM oder PROTEXT erhältlich.

Walter Kriesi, Schweiz/Oberwill

Joyce-Serienbriefe

Das mit dem Joyce PC mitgelieferte Textprogramm LocoScript weist auch in der neuen Version einen erheblichen Mangel auf – Serienbriefe lassen sich nicht ohne weiteres erstellen. Abhilfe schafft hier das Joyce Mailing System, das aus den Programmen Joyce Mail und dem Datei Star besteht. Alle Daten der Dateiverwaltung lassen sich mit LocoScript verarbeiten und ermöglichen eine flexiblere Textgestaltung.
Das Mail-Programm unterstützt des weiteren alle Joyce-Schnittstellen, u.a. kann auf diese Art auch ein Typenraddrucker betrieben werden. Das Mailing System kostet ca. DM 190,–; die Programme Mail und Datei sind aber auch einzeln erhältlich.
Info: Star Division
2120 Lüneburg □

664 Produktion eingestellt

Gemunkelt wurde es schon lange – jetzt ist es offiziell. Mit Datum 1.4.'86 wird die Produktion des CPC 664 eingestellt. Allerdings bedeutet dies für die vielen 664-Besitzer nicht das Aus; weiterhin werden Produkte aus den Bereichen Hard- und Software produziert und angeboten.
Info: Schneider
8939 Türkheim □

Supercalc 2

Von Schneider kommt jetzt Supercalc 2 für alle CP/M Plus Rechner – also für CPC 6128 und Joyce. Das umfangreiche Tabellenkalkulationsprogramm weist hervorragende Features auf und verfügt auch über eine Grafikausgabe. Supercalc 2 liegt eine über 100 Seiten starke Bedienungsanleitung bei, die allerdings in englischer Sprache gehalten ist.
Mit diesem hervorragenden Programm, das auch auf anderen Systemen viele Freunde fand, wurde der Einsatzbereich der Schneider Computer erneut um ein Stück Professionalität erweitert.
Info: Schneider
8939 Türkheim □

Computer mieten

Eine neue Idee präsentiert sich mit der Möglichkeit, Computer zu mieten statt zu kaufen. Diesen Service bietet seit kurzem die Fa. SFK, die Computer und Peripherie auf dieser Basis zur Verfügung stellt.
Info: SFK Electronik
5800 Hagen □

Mathematische Textverarbeitung

Textprogramme gibt es wie Sand am Meer. Oft wünscht man sich jedoch die Einbindung von mathematischen Formeln bzw. Sonderzeichen (z.B. für eine Diplomarbeit), und da fehlt es bei vielen Textprogrammen.
Von GfA kommt nun ein solches mathematisches Textsystem, das allerdings nur in Verbindung mit Epson-kompatiblen Druckern mit Download-Funktion lauffähig ist.
RH-Mathext kostet ca. DM 69,–.
Info: GfA Systemtechnik
4000 Düsseldorf 11 □

Typenraddrucker für Joyce

Jetzt gibt es auch einen Typenraddrucker für den Joyce. Der SD 15 ist sowohl mit paralleler als auch mit serieller Schnittstelle ausgestattet, ein Traktor gehört ebenfalls zum Lieferumfang. Gedruckt wird mit einer Geschwindigkeit von 15 Zeichen pro Sekunde. Per mitgelieferter Software können unter LocoScript erstellte Texte ausgegeben werden. Für ein variables Gestalten des Schriftbildes werden mehrere Typenräder zur Auswahl stehen. Der Preis für den SD 15 beläuft sich auf ca. DM 700,–.
Info: Schneider Data
8050 Freising □

LocoScript-Fibel

Die LocoScript-Fibel ist ein systematisches, einfaches und leicht verständliches Nachschlagewerk, in dem alle Funktionen des Textverarbeitungsprogrammes in Kurzform beschrieben werden, wie man zu ihnen gelangen kann und welche Schritte notwendig sind, um ein konkretes Ergebnis im Einzelfall zu erreichen. Es ist nicht als Ersatz des Handbuches gedacht, vielmehr als nützliche Ergänzung. Durch zusätzliche Verweise auf Kapitel und Seiten des Handbuches können ohne längeres Suchen die einzelnen Punkte zur Vertiefung nachgelesen werden.
Info:
Walter Ehlers
2000 Hamburg 56 □

CPC Music-Box

Das englische Softwarehaus Electric Studio, das bereits für den Lightpen verantwortlich zeichnete, erweiterte seine Angebotspalette um einige interessante Produkte.
Zunächst wird die CPC Music-Box auf den Markt kommen. Dieses Programm ist für alle CPC-Rechner erhältlich und weist hervorragende Leistungsmerkmale auf. Der implementierte Envelope-Designer ist dabei voll Lightpen-kompatibel und kann daher als sehr anwenderfreundlich bezeichnet werden.
Insgesamt lassen sich bis zu 8000 Noten speichern, der CPC wird zum idealen Hilfsmittel für angehende Komponisten.
Info:
Electric Studio
P.O. Box 96, Luton
LU3 2JP □

Do you speak english?

Es gibt jetzt auch einen heißen Draht nach England. Die Firma Amstrad hat aufgrund zahlreicher Anregungen seitens der Anwender reagiert und eine telefonische Kundenberatung bereitgestellt. Für die Telefonbetreuung steht ein geschultes Team zur Verfügung, das alle Fragen während der Geschäftszeiten von Amstrad beantwortet.
Die Amstrad Usersprechstunde ist telefonisch unter folgender Nummer erreichbar:
Tel.: 00-44/0277/230222 □

Joyce-Lightpen

Um die Bedienung des Joyce einfacher zu gestalten, hat sich Electric Studio aus England etwas tolles einfallen lassen. Der mit großem Erfolg vertriebene ES-Lightpen soll in Kürze auch für den Joyce erhältlich sein. Damit wird es möglich sein, dem Anwender eine gute und übersichtliche Benutzeroberfläche für alle denkbaren Programme zu bieten.
Info:
Electric Studio
P.O.Box 96, Luton
Lu3 2JP □

Tasword-Utilities

Tasman Software ist mittlerweile auch hierzulande ein Begriff. Mit den überzeugenden Textverarbeitungsprogrammen Tasword wurde eine Produktlinie geschaffen, die mittlerweile auf nahezu allen gängigen Heimcomputern zum Einsatz kommt. Nach dem erst kürzlich erschienenen Tasword D, das eine Mailmerge-Funktion beinhaltet, gibt es nun das Zusatzprogramm TasSpell.
TasSpell ist ein Programm, das die Syntax der Rechtschreibung überprüft und auftretende Fehler anzeigt. Das Programm hat etwa 20000 Wörter als Vergleichsmöglichkeit gespeichert; evtl. Ergänzungen oder Änderungen sind möglich. TasSpell gibt es zur Zeit nur in Englisch. Bleibt zu hoffen, daß sich schon bald ein deutsches Softwarehaus diesem hochinteressanten Hilfsprogramm annimmt.

Monte-Carlo-Simulation mit LOGO

In der Elektronik und der Physik erfreut sich die Monte-Carlo-Simulation in den letzten Jahren steigender Beliebtheit, weil mit dieser Methode die oft verwickelten und komplizierten Vorgänge in mikrophysikalischen Bereichen theoretisch mit Hilfe des Computers erforscht werden können. Dieser Beitrag soll dem interessierten Nicht-Fachmann an einem allereinfachsten Beispiel ein Gefühl dafür vermitteln, worum es bei der Monte-Carlo-Simulation geht und worin das zugrunde liegende Prinzip besteht.



Foto: Huber, BKV

Es ist eine altbekannte Tatsache, daß man eine Sache dann am besten lernt, wenn man sie selbst tut (Learning by Doing). Daher bietet es sich an, dem interessierten Heimcomputerbesitzer ein Programm an die Hand zu geben, mit dem er selbst einfache Monte-Carlo-Simulationen durchführen kann.
Dabei hat er ausreichend Gelegenheit, durch Modifikationen des Programmes eigene Versuche durchzuführen und die Möglichkeiten der Methode kennenzulernen.
Doch was ist das nun eigentlich, die Monte-Carlo-Simulation? Es hat nichts mit Urlaub, Sonne und Meer zu tun; auch Autorennen interessieren dabei höchstens am Rande; weder gekrönte Häupter spielen eine Rolle noch berühmte Tennisstars; worauf es bei unserem Thema ankommt, ist die Roulettekugel in dem berühmten Spielcasino von Monte Carlo. Nun wird sich mancher fragen, was denn Roulette mit Physik oder Mikroelektronik zu tun hat. Des Rätsels Lösung liegt darin, daß sowohl beim Roulette als auch bei physikalischen Vorgängen kleinster Teilchen der Zufall eine gewichtige Rolle spielt. Genaugenommen ist das nicht nur bei Atomen oder Elektronen der Fall; schließlich sind alle Körper letztlich aus Elementarteilchen aufgebaut und gehorchen damit den gleichen Gesetzen wie diese selbst. Es ist nur so, daß man unter bestimmten Voraussetzungen, wenn eine sehr große Anzahl von Elementarteilchen beteiligt sind, den Zufall nicht mehr merkt, weil die menschlichen Sinne dafür nicht feinfühlig genug sind. Der Zufall mittelt sich sozusagen heraus. Dies trifft zum Beispiel auf die Bewegung der Planeten zu. Seit einigen Jahrhunderten berechnen Menschen ihre Bahn nach stets den gleichen Gesetzen (von einigen Korrekturen durch die Relativitätstheorie einmal abgesehen). Vom Zufall also keine Spur. Es sind jedoch andere physikalische Systeme möglich, bei denen der Zufall, der sich auf allerkleinster Ebene abspielt, verstärkt wird und dadurch auch makroskopisch dem Menschen sichtbar wird.
Ein solches System ist die Roulettekugel (oder auch die 49 Kugeln des Ziehungsgerätes im Deutschen Lottoblock). Selbst wenn es gelänge, durch irgendeine sinnreiche Vorrichtung die Anfangsbedingungen der Ziehung mit der höchsten physikalisch überhaupt möglichen Präzision immer wieder einzustellen, wird man doch bei jedem Versuch ein anderes Ergebnis erhalten. Der Grund dafür ist der, daß der Zufall ein nicht wegzudenkendes Element der grundlegenden physikalischen Gesetze ist.
Die Monte-Carlo-Simulation ist nun eine Methode, das mehr oder weniger zufällige Verhalten solcher physikalischer Systeme auf dem Computer nachzubilden. In der Mikroelektronik gewinnt die Monte-Carlo-Simulation in den letzten Jahren erheblich an Bedeutung, da durch die immer zunehmende Miniaturisierung der Schaltkreise die Bauelemente (Transistoren etc.) so klein werden, daß die herkömmlichen makroskopischen Gleichungen nicht mehr gültig sind; die quantenmechanischen Gesetze werden hier immer stärker spürbar. Um das Verhalten der Elektronen im Halbleiter unter diesen Bedingungen zu erforschen, simuliert man die Bewegung von beispielsweise 50000 Elektronen mit Hilfe der Monte-

Carlo-Simulation, um etwa zu berechnen, welcher Strom bei einer bestimmten Spannung fließt.

Die LOGO-Schildkröte in Monte-Carlo

Für unser einfaches Beispiel wollen wir uns jedoch damit begnügen, das Verhalten eines einzelnen Elektrons zu simulieren. Wem es lieber ist, der kann sich auch vorstellen, daß es sich um ein Gasmolekül in einem Gasvolumen handelt; die physikalischen Gesetze sind, soweit es jedenfalls unser einfaches Modell betrifft, identisch.

Als Programmiersprache für unsere Simulation wählen wir die Sprache Logo, weil Logo einerseits einen eingebauten Zufallszahlengenerator besitzt, der das Herz jeder Monte-Carlo-Simulation ist; andererseits bietet die Turtle-Grafik eine hervorragende Möglichkeit, das dynamische Verhalten direkt im Bild darzustellen.

Die Logo-Schildkröte stellt also in unserem Fall das Elektron (oder Gasmolekül) dar. Die simulierten Bewegungen des Elektrons werden direkt durch Bewegungen der Schildkröte auf dem Schirm des Computers nachgebildet, so daß der Betrachter einen unmittelbaren Eindruck des Verhaltens gewinnt. Nun sind noch Regeln aufzustellen, nach denen sich unsere Schildkröte bewegen soll.

Das Verhalten von Elektronen im Festkörper und von Molekülen eines Gases ist im wesentlichen durch drei Effekte bedingt:

1. Die Massenträgheit des Teilchens (ohne einwirkende Kräfte fliegt es einfach geradeaus);
2. Die zufälligen Zusammenstöße mit irgendwelchen Stoßpartnern (Atome des Halbleiterkristalls beim Elektron, andere Gasteilchen beim Gasmolekül);
3. Die Einwirkung von Kraftfeldern (elektrisches Feld beim Elektron, Schwerkraft beim Gasmolekül).

Nun soll Schritt für Schritt ein Simulationsprogramm entwickelt werden, in dem bei jedem Schritt ein weiterer der oben genannten Effekte hinzukommt.

Die träge Schildkröte

Beginnen wir mit der Massenträgheit. Um den zeitlichen Verlauf unserer Simulation zu berechnen, teilen wir den Zeitablauf in viele kleine Abschnitte :dt auf (Diskretisierung). Für jeden Zeitpunkt am Ende eines solchen Abschnitts wird der Zustand unseres Teilchens (bzw. unserer Schildkröte) berechnet und auf dem Schirm dargestellt. Der Zustand des Teilchens ist zu jedem Zeitpunkt gegeben durch:

- den Ort, gekennzeichnet durch die Position auf dem Bildschirm;
- den Betrag der Geschwindigkeit, den wir uns in einer Variablen :v merken, und
- die Richtung der Geschwindigkeit, die durch den Winkel :phi dargestellt wird (s. Bild 1). Damit dieser Winkel mit dem Bild auf dem Schirm übereinstimmt, muß dafür gesorgt werden, daß die Schildkröte auch immer tatsächlich in diese Richtung „sieht".

Nun können wir anfangen, unser Hauptprogramm zu schreiben (s. Tabelle 1). Es trägt den Namen "mc" für "Monte-Carlo"; Eingabeparameter sind die Zeiten :ta und :te, die die Zeitpunkte von Anfang und Ende unserer Simulation angeben. Der erste Teil des Hauptprogrammes besteht aus Initialisierungen. Zunächst löschen wir den Schirm und geben das "wrap"-Kommando, damit uns die Schildkröte nicht auf Nimmerwiedersehen jenseits des Bildschirmrandes verschwindet. Dann werden Betrag :v und Richtung :phi der Geschwindigkeit auf die Anfangswerte :v0 und :phi0 initialisiert. Die Anfangsgeschwindigkeit steht in direktem Zusammenhang mit der Temperatur der Umgebung: je höher die

Tabelle 1
Simulation des kräftefreien Teilchens

```
to mc :ta :te
ts
ct
pr [Monte-Carlo Simulation]
pr [(C) 1986 Dr.-Ing. D. Schröder]

wrap

make "v :v0
make "phi :phi0
seth 90 - :phi
make "t :ta
make "dt 1

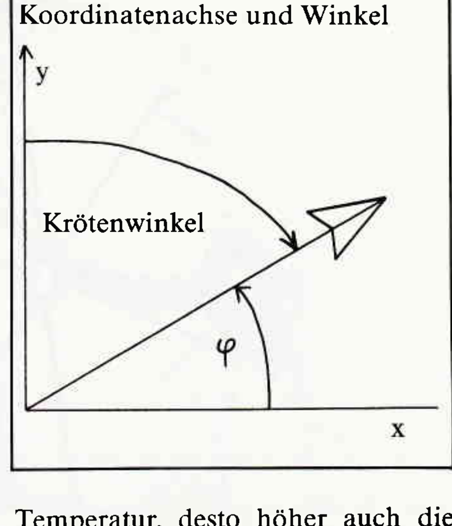
repeat (:te - :ta) / :dt [zeitschritt]
end


to zeitschritt
make "t :t + :dt
fd :v * :dt
end


make "phi0 230
make "v0 20


mc 0 1000
```

Bild 1
Koordinatenachse und Winkel



Temperatur, desto höher auch die (thermische) Geschwindigkeit des Teilchens (thermische Wimmelbewegung). Der nachfolgende "seth"-Befehl dreht die Schildkröte in die Richtung von :phi. Die zusätzliche Differenz von 90 Grad kommt gemäß Bild 1 daher, daß :phi entsprechend den in der Physik üblichen Konventionen von der waagerechten x-Achse links herum gezählt wird, während die Schildkrötenrichtung in Logo von der senkrechten Achse nach rechts im Uhrzeigersinn zählt. Schließlich wird die aktuelle Zeit :t auf die Anfangszeit :ta und der Zeitschritt :dt willkürlich auf 1 gesetzt.

Damit ist die Initialisierung abgeschlossen. Nun folgt die eigentliche Simulation. Hier machen wir es uns ganz einfach, indem wir uns um die lästigen Einzelheiten gar nicht erst kümmern: Es wird so oft, wie der Zeitschritt in die Simulation paßt ein Unterprogramm "zeitschritt ausgeführt, in dem alles weitere erledigt wird. Das wäre für das Hauptprogramm alles.

Wenn wir nun unser Programm starten, teilt uns der Logo-Interpreter mit der Fehlermeldung "I don't know how to zeitschritt" mit, daß er sich um die lästigen Einzelheiten auch nicht kümmern will. Also bleibt es schließlich doch an uns hängen, in diesem Punkt konkret zu werden.

Aber es gibt Schlimmeres. Fürs Erste beschränken wir uns auf den simplen Fall, daß sich das Teilchen entsprechend der Massenträgheit mit gleichbleibender Geschwindigkeit weiterbewegt, da ja vorerst keine Kräfte wirken sollen. Das "zeitschritt-Programm nimmt damit die einfache Gestalt laut Tabelle 1 an. Zunächst treiben wir etwas Buchführung und zählen unsere „Uhr", die Variable :t, um den Zeitschritt :dt weiter. Dann bewegt sich die Schildkröte gemäß der

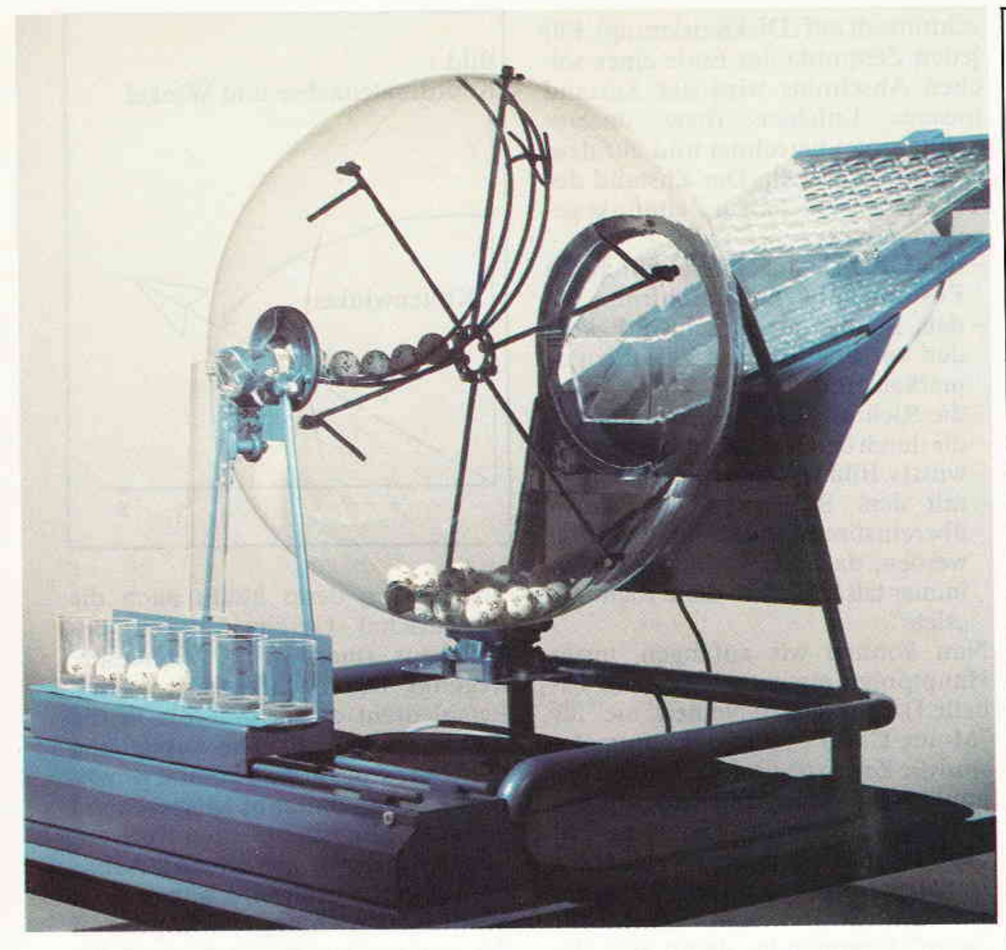


Foto: Lotterie-Treuhand-GmbH

allen aus der Schule bekannten Formel, daß der zurückgelegte Weg gleich Geschwindigkeit mal Zeit ist, entsprechend :v*:dt nach vorne. Und das wäre für den ersten Fall alles. Schließlich müssen wir mittels des "make"-Befehls den Anfangswerten :v0 und :phi0 noch Werte zuweisen und können dann unser Programm mit

mc 0 1000

starten. Die Schildkröte zieht nun auf einer geraden Linie ihres Weges, wie es sich für einen trägen Körper gehört. Wir bemerken, daß sie manchmal innehält, als ob sie überlegen wollte, wie es weitergeht. Das ist aber eine Eigenart des Logo-Interpreters, der von Zeit zu Zeit seinen Speicher wieder aufräumen muß.

Die angerempelte Schildkröte

Im nächsten Schritt wollen wir die Streuungen des Elektrons an den Atomen des Halbleiters berücksichtigen. Von Zeit zu Zeit erleidet das Elektron Zusammenstöße mit dem Kristallgitter des Halbleitermaterials, bei denen es in eine zufällige Richtung abgelenkt wird. Je nach Stoßmechanismus kann es dabei Energie mit dem Kristallgitter austauschen, so daß es nach der Streuung wieder die thermsche Geschwindigkeit :v0 besitzt (wird erst nachher interessant, wenn das Elektron durch ein elektrisches Feld zwischen den Stößen beschleunigt wird). Dieser Effekt ist dafür verantwortlich, daß sich ein von einem starken Strom durchflossener Draht erhitzt (Elektroherd). Es gibt aber auch Stoßmechanismen, bei denen die Energie des Elektrons erhalten bleibt. In diesem Fall spricht man von einer Aufheizung der Elektronen. Wir wollen uns hier aber auf den ersten Fall beschränken.

Um die Streuung zu simulieren, fügen wir die Ergänzung laut Tabelle 2 in unser Programm ein. Dazu definieren wir ein Unterprogramm namens "streu, das entsprechendes leistet. Hier wird also zunächst die Geschwindigkeit auf den Anfangswert :v0 gesetzt. Anschließend erfolgt die Ablenkung in eine zufällige Richtung, die wir mit Hilfe von "random 360" erzeugen. Schließlich muß die Schildkröte auf dem Schirm in die neue Richtung gedreht werden, wozu wir wieder den "seth"-Befehl benutzen.

Nun müssen wir diesen Streumechanismus in unser Zeitschritt-Programm einbauen. Dabei ist zu beachten, daß die Stöße des Teilchens nicht in gleichen Zeitabständen, sondern zufällig erfolgen. Die Häufigkeit der Streuungen ist durch die Streuwahrscheinlichkeit pro Zeitschritt gegeben, die wir uns in der Variablen :wstreu merken. Wir geben die Streuwahrscheinlichkeit in Prozent, das heißt zwischen 0 und 100, an. Eine Wahrscheinlichkeit von 0 bedeutet, daß das Teilchen niemals gestreut wird; 100 heißt Streuung bei jedem Zeitschritt, und 50 bedeutet eine Streuung bei im Mittel jedem zweiten Zeitschritt. Es muß jetzt im Unterprogramm "zeitschritt dafür gesorgt werden, daß entsprechend der Streuwahrscheinlichkeit unser Streu-Programm aufgerufen wird. Dazu erzeugen wir uns eine Zufallszahl zwischen 0 und 100 und vergleichen sie mit der Streuwahrscheinlichkeit. Das Programm "streu wird dann nur aufgerufen, wenn die Zufallszahl kleiner als die Wahrscheinlichkeit ist. Dadurch ist gewährleistet, daß die Streuung im Mittel so oft vorkommt, wie es die Streuwahrscheinlichkeit angibt.

Tabelle 2
Simulation des gestreuten Teilchens
(Nur veränderte oder neue Routinen angeben. ! bedeutet "über Zeilenende hinwegschreiben".)

```
to mc :ta :te
ts
ct
pr [Monte-Carlo Simulation]
pr [(C) 1986 Dr.-Ing. D. Schröder]

repeat 5 [pr []]
eing [Streuwahrscheinlichkeit [0 - 100]!
 ?] "wstreu

wrap

make "v :v0
make "phi :phi0
seth 90 - :phi
make "t :ta
make "dt 1

repeat (:te - :ta) / :dt [zeitschritt]
end


to zeitschritt
make "t :t + :dt
fd :v * :dt

if (random 100 < :wstreu) [streu]
end

to streu
make "v :v0
make "phi random 360
seth 90 - :phi
end

to eing :text :wert
type :text
make :wert rq + 0
end
```

Schließlich bauen wir am Anfang in unser Hauptprogramm noch eine Eingabe der Streuwahrscheinlichkeit ein. Dazu definieren wir ein Unterprogramm namens "eing", das einen entsprechenden Abfragetext, der als Parameter :text übergeben wird, auf dem Bildschirm ausgibt und dann auf eine Benutzereingabe wartet. Der Benutzer gibt nun eine Zahl ein, die der Variablen, die durch den zweiten Parameter :wert festgelegt ist, zugewiesen wird.
Hierbei ist noch eine Besonderheit des Schneider-Logo zu beachten. Die Logo-Funktion "rq" liest ein Wort von der Tastatur ein und übergibt es mit vorangestellten Anführungszeichen an das aufrufende Programm. Bei Eingabe einer Zahl – z.B. 100 – wird diese als "100 übergeben; dieser Ausdruck stellt aber für den Vergleich keine Zahl, sondern ein Wort dar (es können auch Wörter in Logo miteinander verglichen werden; der Vergleich erfolgt dann lexikalisch). Deshalb müssen wir nach einer Möglichkeit suchen, um "100 in 100 zu verwandeln. Glücklicherweise behandelt der Logo-Interpreter "100 in arithmetischen Ausdrücken als Zahl. Daher genügt es, "100 + 0 zu schreiben, um 100 zu erhalten. Das ist der Grund für den etwas merkwürdigen Ausdruck in der letzten Zeile des "eing"-Programmes.
Nachdem wir mit Hilfe dieses Unterprogrammes eine Abfrage der Streuwahrscheinlichkeit am Anfang unseres Hauptprogrammes eingebaut haben, können wir die Simulation erneut starten. Bei Eingabe einer Streuwahrscheinlichkeit von 0 bleibt alles beim alten; wählen wir jedoch einen Wert von 40 oder gar 100, wird unser simuliertes Teilchen entsprechend oft einen imaginären Stoß erleiden, der es in eine andere Richtung drängt.
Damit haben wir – zumindest im Prinzip – das Verhalten eines Elektrons im Halbleiterkristall ohne elektrisches Feld oder des Gasmoleküls, wenn man die Schwerkraft vernachlässigen kann (z.B. bei sehr leichten Molekülen), simuliert. Man sieht, daß unser Teilchen, ausgehend von seinem Ursprungsort, nach und nach alle Bereiche des Bildschirms durchwandert. Mit diesem Bild vor Augen fällt es leicht sich vorzustellen, daß eine große Zahl von Teilchen sich vom Ursprungsort aus langsam über das gesamte Gebiet verteilen würde; dieser Effekt wird Diffusion genannt. Er ist z.B. dafür verantwortlich, daß die Duftmoleküle eines Raumsprays sich nach und nach im ganzen Zimmer verteilen. Die Funktion des Transistors beruht ebenfalls auf Diffusion; sie sorgt dafür, daß die Elektronen aus der Basis trotz der sperrenden Kollektor-Basis-Diode in den Kollektor gelangen können.

Die beschleunigte Schildkröte

Nun wollen wir den dritten der oben angeführten Effekte in unsere Simulation einbauen: die Einwirkung eines Kraftfeldes (s. Tabelle 3).
Analog zu unserem Vorgehen bei der Streuung definieren wir uns ein Unterprogramm "feld, das die Wirkung des Kraftfeldes während eines Zeitschrittes beschreibt. Diese besteht darin, daß das Teilchen in Richtung der Kraft beschleunigt wird. Wir müssen also bestimmen, wie sich die Geschwindigkeit und die Richtung während eines Zeitschrittes ändern. Dabei kann man qualitativ sagen, daß der Betrag der Geschwindigkeit größer werden muß, wenn sich das Teilchen ungefähr in Kraftrichtung bewegt; anderenfalls wird es abgebremst. Außerdem wird die Richtung der Geschwindigkeit in die Kraftrichtung gedreht. Die genauen Formeln und, für mathematisch interessierte Leser, die Herleitung sind in Tabelle 4 angegeben.
Entsprechend der Formeln berechnen wir also für jeden Zeitschritt die Änderung der Geschwindigkeit :dv und die Änderung der Richtung :dphi; diese werden dann zu den bisherigen Werten addiert. Außerdem wird mit "seth" wieder die Schildkröte in die neue Richtung gedreht. Die Masse unseres Teilchens tritt übrigens in unserer Simulation nicht explizit auf; wir haben sie der Einfachheit halber auf 1 gesetzt. Bei der Berechnung von :dphi dürfen wir außerdem eine Umrechnung von Bogenmaß in Grad nicht vergessen; dazu dient der Faktor 180/pi.
Der Aufruf des Unterprogrammes "feld wird nun einfach in das Programm "zeitschritt eingebaut. Weiterhin fügen wir ins Hauptprogramm mit Hilfe von "eing eine Abfrage der Kraft ein. Dabei beschränken wir uns in unserem Beispiel auf eine Kraft von x-Richtung, die Komponente :fy in y-Richtung setzen wir 0. Schließlich stellen wir noch in einer kleinen Funktion namens "pi die gleichnamige Zahl zur Verfügung.
Damit haben wir alles für eine komplette Simulation zusammen. Nach Start des Programmes können wir zwischen verschiedenen Möglichkeiten wählen. Wenn wir die Kraft auf Null setzen, erhalten wir ein Verhalten wie im vorigen Abschnitt. Geben wir für die Kraft z.B. 5 ein und setzen die Streuwahrscheinlichkeit auf 0, zeigen sich die, manchen vielleicht aus der Schule bekannten, Wurfparabeln. Wählen wir beide Parameter ungleich Null, sehen wir das qualitative Verhalten des Elektrons im Halbleiter auf dem Bildschirm unseres Computers, das aus einer Folge von Beschleunigungen durch das Kraftfeld, immer wieder unterbrochen von Stoßvorgängen, besteht.

Tabelle 3
Simulation mit Streuung und Feld
(nur veränderte oder neue Routinen
! bedeutet "über Zeilenende hinwegschreiben.)

```
to mc :ta :te
ts
ct
pr [Monte-Carlo Simulation]
pr [(C) 1986 Dr.-Ing. D. Schröder]

repeat 5 [pr []]
eing [Kraft in x - Richtung [0 - 10] ?]!
 "fx
pr []
eing [Streuwahrscheinlichkeit [0 - 100]!
 ?] "wstreu

wrap

make "v :v0
make "phi :phi0
seth 90 - :phi
make "t :ta
make "dt 1

repeat (:te - :ta) / :dt [zeitschritt]
end


to zeitschritt
make "t :t + :dt
fd :v * :dt

feld
if (random 100 < :wstreu) [streu]
end


to feld
(local "dv "dphi)
make "dv ((:fx * cos :phi) + (:fy * sin!
 :phi)) * :dt
make "v :v + :dv
make "dphi ((:fy * cos :phi) - (:fx * s!
in :phi)) / :v * :dt * 180 / pi
make "phi :phi + :dphi
seth 90 - :phi
end


to pi
op 3.14159
end


make "fy 0
```

Ausblick

Hiermit sind wir am Ende unserer kleinen Einführung in die Monte-

Carlo-Simulation angelangt. Es konnte nur anhand eines einfachen Beispiels ein kleiner Einblick in die Thematik gegeben werden. In der Forschung sind die untersuchten Phänomene noch weitaus komplexer. Im Halbleiter treten beispielsweise verschiedene Streumechanismen auf, die von der Energie und der Richtung abhängen; auch das Kraftgesetz ist nicht mehr so einfach wie in unserem Beispiel.

Der interessierte Leser sei dazu aufgerufen, eigene Experimente mit diesem Programm zu machen; dazu bietet sich eine Fülle von Möglichkeiten. Beispielsweise hat man in der Praxis wenig davon, wenn man sich das Verhalten der Teilchen nur auf dem Bildschirm ansieht. Hier wird die Methode dazu eingesetzt, um bestimmte Materialeigenschaften des Halbleiters zu berechnen. Eine wichtige Größe ist etwa die mittlere Geschwindigkeit der Elektronen in Abhängigkeit von der Stärke des elektrischen Feldes. Diese Größe bestimmt u.a. die Schnelligkeit, mit der Schaltvorgänge in Transistoren ausgeführt werden. Man könnte nun in unsere Simulation ein Unterprogramm einbauen, das bei jedem Zeitschritt diesen Mittelwert auf den Bildschirm schreibt, um zu erforschen, wie die mittlere Geschwindigkeit von der Stärke des Kraftfeldes abhängt.

Eine weitere Möglichkeit wäre, entsprechend den Formeln in Tabelle 4, die Masse des Teilchens zu berücksichtigen und mit verschiedenen Massen zu experimentieren.

Der fortgeschrittene Leser könnte einmal ausprobieren, was passiert, wenn statt des elektrischen, ein magnetisches Feld auf das Elektron wirkt. Wenn das magnetische Feld senkrecht zur xy-Ebene steht, müßten sich unter bestimmten Bedingungen Kreisbahnen ergeben.

Bisher war unser Kraftfeld immer konstant. Man könnte untersuchen, was bei einer zeitlich veränderlichen Kraft passiert. Dazu würde man eine Funktion "fx :t einführen, die den Wert der Kraft in x-Richtung zum Zeitpunkt :t liefert.

Etwas fortgeschrittenere Logo-Kenntnisse würde die Simulation der Wechselwirkung von zwei Teilchen erfordern, bei der man zwischen zwei Schildkröten hin und her schalten muß.

Dies waren nur einige Beispiele, in welcher Richtung man hier noch weiter vordringen kann; der Phantasie sind da keine Grenzen gesetzt.

Schließlich möchte der Autor der Hoffnung Ausdruck verleihen, daß er mit diesem Artikel ein wenig das Interesse an der Monte-Carlo-Simulation und der dahinterstehenden Physik geweckt hat. Er wünscht viel Vergnügen beim weiteren Experimentieren und beim Betrachten der Bilder, die manchmal eine leise Ahnung davon vermitteln können, wie Ordnung aus dem Chaos entsteht.

(Dr.Ing. Dietmar Schröder)

Tabelle 4

Die Wirkung des Kraftfeldes

Herleitung der im Unterprogramm "feld verwendeten Formeln ausgehend von Newtons Gesetz (Kraft = Masse x Beschleunigung)

$$\vec{F} = m\,\vec{a} \qquad (1)$$

und der Definition "Beschleunigung = Änderung der Geschwindigkeit/Zeitintervall":

$$\vec{a} = \frac{d\vec{v}}{dt} \qquad (2)$$

erhält man

$$\frac{d\vec{v}}{dt} = \frac{1}{m}\vec{F} \qquad (3)$$

bzw. in Komponentenschreibweise

$$\frac{dv_x}{dt} = \frac{1}{m}F_x \qquad (3a)$$

$$\frac{dv_y}{dt} = \frac{1}{m}F_y \qquad (3b)$$

Für unser Logo-Programm brauchen wir jedoch nicht x- und y-Komponenten der Geschwindigkeit, sondern Betrag v und Richtung gemäß Bild 1. Der Betrag der Geschwindigkeit ist durch

$$v = \sqrt{v_x^2 + v_y^2} \qquad (4)$$

und die Richtung durch

$$v_x = v \cos\varphi \qquad (5a)$$

$$v_y = v \sin\varphi \quad \text{gegeben.} \qquad (5b)$$

Die zeitliche Änderung des Betrages ist durch Differentiation von (4) nach der Kettenregel

$$\frac{dv}{dt} = \frac{1}{2v}\left(2v_x\frac{dv_x}{dt} + 2v_y\frac{dv_y}{dt}\right)$$

woraus mit (3a) und (3b)

$$\frac{dv}{dt} = \frac{1}{m}\left(\frac{v_x}{v}F_x + \frac{v_y}{v}F_y\right)$$

und mit (5a) und (5b)

$$\frac{dv}{dt} = \frac{1}{m}\left[F_x\cos\varphi + F_y\sin\varphi\right] \quad \text{folgt.} \qquad (6)$$

Die Änderung des Richtungswinkels erhalten wir durch Ableitung von (5a):

$$\frac{dv_x}{dt} = \frac{dv}{dt}\cos\varphi - v\frac{d\varphi}{dt}\sin\varphi$$

wobei wir die Ableitungsregeln für Sinus und Cosinus sowie die Kettenregel benutzt haben. Einsetzen von (3a) und (6) liefert

$$\frac{1}{m}F_x = \frac{1}{m}\left[F_x\cos\varphi + F_y\sin\varphi\right]\cos\varphi - v\frac{d\varphi}{dt}\sin\varphi$$

Löst man diese Beziehung nach $\frac{d\varphi}{dt}$ auf, erhält man

Mit $\sin^2\varphi + \cos^2\varphi = 1$ folgt schließlich

$$\frac{d\varphi}{dt} = \frac{1}{m\,v}\left[-F_x\sin\varphi + F_y\cos\varphi\right] \qquad (7)$$

Mit Hilfe der hier abgeleiteten Gleichungen (6) und (7) werden die Änderungen der Geschwindigkeit und des Winkels, "dv und "dphi in der Routine "feld in Tabelle 3 berechnet.

Schneider CPC Software-Schnell-Versand



Diskette 3"

Titel	Preis
Airwolf	47,90
Alien Highway	47,90
Bomb Jack	47,90
Bruce Lee	47,90
Combat Lynx	47,90
Crafton + Xunk	47,90
Cyrus 3D Schach	47,90
Deathville	47,90
Devils Crown	47,90
Elite deutsch	64,90
Fighter Pilot	47,90
Fighting Warrior/ Exploding Fist(1 Disc)	69,00
Frank Bruno Boxing	47,90
Friday the 13th	39,90
Goonies	47,90
Hacker	47,90
Hyper Sports	47,90
Lords of Midnight	47,90
Mindshadow	47,90
Rocky Horror Show	47,90
Space Invasion	47,90
Spindizzy	47,90
Spitfire 40	47,90
Sorcery Plus	47,90
Staicway to Hell	47,90
Strangeloop	47,90
Sweevos World	47,90
Tau Ceti	47,90
Theatre Europa	54,00
Tornado Low Level	47,90
Way of the Tiger	47,90
Who dares wins II	47,90
Winter Games	47,90
Winter Sports	47,90
Wizards Lair	47,90
Yie ar Kung Fu	44,90
3 D Grand Prix	47,90
2112 AD	49,00

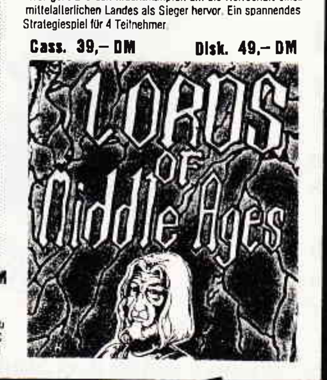


Cass. 39,– DM Disk. 49,– DM

Das Super-Tool. Fünfzig neue Befehle für Grafik, Sound und Programmierung. Diese Basicerweiterung bringt Ihren CPC 464 auf Trab. Bitte lesen Sie die Review in CPC International

Wer geht aus den Machtkämpfen um die Herrschaft eines mittelalterlichen Landes als Sieger hervor. Ein spannendes Strategiespiel für 4 Teilnehmer.

Cass. 39,– DM Disk. 49,– DM





Kassette

Titel	Preis
Airwolf	29,00
Alien Highway	35,90
Bruce Lee	24,90
Combat Lynx	29,90
Crafton + Xunk	36,90
Cyrus 3D Schach	33,90
Daley Thompson Super Test	34,90
Devils Crown	29,00
Doomsday Blues	34,90
Dragontorc	29,00
Elite deutsch	52,90
Fighter Pilot	29,00
Fighting Warrior	29,00
Frank Bruno Boxing	29,00
Hacker	33,90
Hexenkueche	29,00
Hyper Sports	29,00
I, of the Mask	37,90
Laser Basic	59,00
Lord of the Rings	59,00
Lords of Midnight	34,00
Mindshadow	37,00
Monopoly	34,90
N.O.M.A.D	29,00
Ping Pong	29,00
Rambo II	29,90
Saboteur	34,90
Space Invasion	33,90
Spindizzy	36,90
Spitfire 40	34,90
Strangeloop	29,90
Tau Ceti	33,90
Theatre Europa	38,90
The Way of the Tiger	36,90
The Goonies	33,90
They sold a million	36,90
They sold a million 2	39,00
Tornado Low Level	29,00
Turbo Esprit	29,90
Way of the Tiger	37,90
Winter Games	33,90
Winter Sports	36,90
Yie ar Kung Fu	29,00

FRANKIE

Artikel	Beschreibung	Preis
Rushware Mouse für CPC, C64, Atari		148,-
256k Silicon Disc für CPC's		379,-
Vortex F1-S	5,25" Einfachlaufwerk, 708 KByte, Controller, CP/M	1198,–
Vortex F1-D	5,25" Doppellaufwerk, 1,4 MByte, Controller, CP/M	1698,–
Vortex M1-S	wie F1-S, jedoch in 3,5"	998,–
Vortex M1-D	wie F1-D, jedoch in 3,5"	1498,–
Vortex WD-10	10 MByte Festplatte mit umfangreichen Utilities	2689,–
Vortex WD-20	20 MByte Festplatte mit umfangreichen Utilities	3289,–
Vortex F1-X	5,25" Zweitlaufwerk, 708 KByte, ROM-Software	759,–
Vortex M1-X	wie F1-X, jedoch in 3,5"	759,–

Speichererweiterungen für CPC 464/664

Speichererweiterung	Preis	Erweiterungssätze	Preis
SP 64	275,–		
SP 128	348,–		
SP 256	478,–	KIT 64	73,–
SP 320	528,–	KIT 256	213,–
SP 512	589,–	KIT 512	314,–

dk'tronics 64K Erweiterung 179,–
erlaubt die Benutzung von CP/M 3.0 auf dem 464/664, mit Basic-Erweiterung, CPC 6128 – Emulator und Handbuch

Dataphon S21d Komplettset mit Kabel + Software für 664/6128 398,–

Softwareautoren aufgepasst !!!

Wir suchen dringend weitere Programme für unseren Softwarevertrieb. Bitte kurz schriftlich oder telefonisch Kontakt aufnehmen !

Professionelle CP/M Software

Programm	Preis	
WordStar	199,–	(xxxx)
dBase II	199,–	(xxxx)
Multiplan	199,–	(xxxx)
Small C	148,–	(xxxo)
Turbo Pascal	225,–	(xxxx)
Turbo Pascal mit Graphik	285,–	(xxxo)
Turbo Tutor deutsch	104,–	(xxxx)
Turbo Toolbox	225,–	(xxxx)
Turbo Graphik Toolbox	225,–	(xxxo)
Turbo Lader Grundpaket	138,–	(xxxx)
Turbo Lader Business	148,–	(xxxx)
Turbo Lader Science	198,–	(xxxx)
Star Writer I	198,–	(xxxo)
Datei Star	98,–	(xxxo)
Business Pack	198,–	(ooox)

Bei jeder Bestellung unbedingt Computertyp angeben!



Kassette

Titel	Preis
Tasword Kassette	49,00
Tasword Diskette	69,00
Tasword 6128 Diskette	96,00
Statistic - Star Kass	59,00
Statistic - Star Disc	79,00
Star - Mon Kassette	59,00
Star - Mon Diskette	79,00
Disksort - Star Disc	49,00
Star - Mail(JOCYE) Disc	98,00
Pascal1/MT+ Disc	174,00
CBasic Compiler Disc	174,00
DR Draw Disc	199,00
DR Graph Disc	199,00

Video-Digitizer 1000 incl. Software für alle CPC's 395,–

Auf Postkarte kleben oder in Briefkuvert stecken und ab geht die Post!

Ja, schicken Sie mir umgehend folgende Artikel aus Ihrem Angebot

○ V-Scheck anbei ○ per Nachnahme (zzgl. 5,- DM Versandkosten) ○ Bargeld anbei

___ Stück ___ DM
___ Stück ___ DM
___ Stück ___ DM
___ Stück ___ DM

Versandkosten ___ DM
Gesamtsumme ___ DM

Absender (deutlich schreiben)

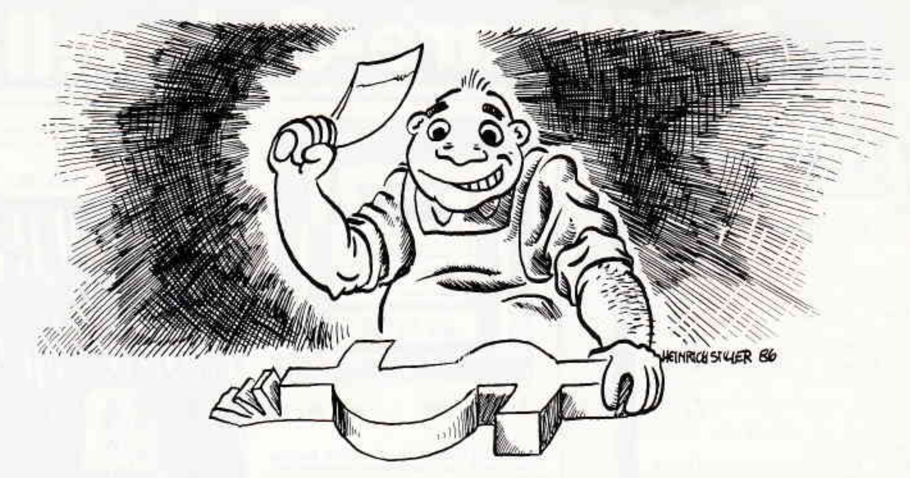
Heimcomputer-Shop

Waldeck-Automaten Vertriebsgesellschaft mbH

Bahnhofstraße 10
2870 Delmenhorst

Telefon (04221) 16464

String-verarbeitung – ein Problem?



Stringverarbeitung, das wäre doch ein Thema, denke ich mir und lege auch sofort los.
Schon nach kurzer Zeit fällt mir auch schon das erste Problem ins Auge. So gut, wie ich dachte, weiß ich bei der Stringverarbeitung gar nicht Bescheid. Doch so schnell gibt unser "Papa seinen Sohn" nicht auf, denn wie schon der Dichter sagt: Schlag nach bei Shakespeare...!

Aber sehen wir doch erst mal, was der kleine, graue Klumpen, den manche Leute als Gehirn bezeichnen, selber noch weiß.
Als String bezeichnet man eine Folge von Buchstaben, Zahlen und anderen Zeichen, bis zu einer maximalen Länge von 255 Zeichen.
Der eigentliche String muß von Anführungszeichen eingeschlossen sein. Stringverknüpfungen funktionieren mit einem normalen "+"-Zeichen. Was vermittelt uns nun das CPC-Handbuch zu diesem Thema?
Wie zu vielen anderen Themen, wird auch hier alles sehr allgemein gehalten. Es gibt ein paar simple Beispiele, drei bis vier Sätze (die kaum etwas Neues bringen) und natürlich die meisten Basic-Kommandos zur Stringverarbeitung.
Zur Auftrennung der Strings gibt es da die LEFT$-, MID$- und RIGHT$-Befehle. Wollen Sie z.B. eine Reihe von Dollarzeichen auf den Bildschirm bringen, vielleicht als Umrahmung eines Menues, so benutzen Sie doch einfach den STRING$-Befehl.
Er gibt ein Zeichen sooft wieder, wie die in Klammern angegebene Zahl es vorschreibt. Hierdurch sparen Sie, z.B. in Mode 2, die Eingabe von 80 Zeichen. Dieses sieht dann folgendermaßen aus:
10 PRINT STRING$ (80,"$")
Da wohl jeder CPC-User auch ein Handbuch besitzt, ist es wohl überflüssig, alle Befehle hier noch einmal zu erklären – schauen Sie doch selber nach.
So viel zum Handbuch, aber besonders schlau war ich nun immer noch nicht. Doch dann ging mir ein Kronleuchter auf (Jubel) und ich fragte einfach meinen Kumpel Igor – einen sehr erfahrenen Operator.
Kaum daß ich gefragt hatte, sprudelte es auch schon drei Stunden am Stück aus Igor heraus. Da ich beim besten Willen nicht alles mitschreiben konnte, möchte ich Ihnen hier nur eine Zusammenfassung geben.
Wenn man von der maximalen Länge der Strings absieht, gibt es grundsätzlich keine größeren Probleme bei der Stringverarbeitung. Kommt man jedoch mit umfangreicher Stringverarbeitung (z.B. in einem Textprogramm) in Berührung, so erblickt das trübe Auge einen merkwürdigen Effekt. Es tritt ein oft Minuten dauerndes, stehenbleibendes Programm auf.
In dieser Zeit scheint der Compi mit den Gedanken ganz woanders zu sein, denn Eingaben irgendwelcher Art sind völlig sinnlos und zeigen absolut keinen Erfolg.
Diesen Zustand nennt man "Garbage Collection", was ungefähr mit Müllbeseitigung zu übersetzen wäre. „Was hat denn die Stringverarbeitung mit Müllbeseitigung zu tun?" werden Sie fragen. Das habe ich Igor auch gefragt und der war, wie immer, um eine Antwort nicht verlegen.
Der Speicher des CPC's stellt für Strings einen bestimmten Bereich zur Verfügung. Dieser Bereich beginnt bei &B08E und endet bei &B08F, &B090.
Bei jeder Änderung eines Strings bleibt der alte ungültig zurück. Diese ungültigen Strings sind "Garbage", also Müll.
Ist nun der Speicherbereich für Strings soweit gefüllt, daß ein nun anstehender String nicht mehr hinein paßt, leitet der Rechner die "Garbage Collection" ein.
Das bedeutet, der Rechner läuft den gesamten Speicherbereich ab und sortiert alle gültigen Strings der Reihe nach, von der höchsten Speicheradresse her, ein. Ist der letzte gültige String einsortiert, ist alles nach diesem bis zum Ende des Speicherbereiches als Müll erkannt und kann beseitigt werden. Im Speicher ist neuer Platz geschaffen und die eigentliche Stringverarbeitung kann weiter gehen.
Die Dauer der "Garbage Collection" ist nur von der Anzahl der Strings abhängig, kann also nicht verkürzt werden.
„Was soll ich den nun tun, um diesen zeitraubenden Vorgang zu verhindern oder zumindest so selten wie möglich auftreten zu lassen?" werden Sie fragen.
Ich habe auch gefragt, nämlich meinen Kumpel Igor.
Der schien ziemlich verstört zu sein, daß ich nicht selber darauf kommen konnte, aber dann hat er mir doch geholfen. „Ganz einfach!" meinte er. „So wenig Stringmüll wie möglich produzieren".
Igor baut zu diesem Zweck einen neuen String in einen String, der nicht mehr benutzt wird, ein.
Dies macht er mit einer besonderen Form des MID$-Befehls:
10 a$="Duebelloch"
20 MID$=(a$,3,5)="eisen"
Der String lautet nun:
"Dueisennoch"
Die einzelnen Teile dieses Strings lassen sich durch die, anfangs schon erwähnten, Teilungsbefehle wieder trennen und verarbeiten.
Durch diese Maßnahme spart man Strings und damit auch Müll. Ganz verhindern läßt sich die "Müllabfuhr" des CPC's damit, zumindest in sehr ausführlichen Programmen, wahrscheinlich nicht, doch tritt sie wesentlich seltener auf.
Wollen Sie die Garbage-Collection ganz vermeiden, so wenden Sie den Print Free (" ")-Befehl an. Er löst die Garbage-Collection aus, und zwar dann, wenn Sie es wollen. Hierdurch halten Sie die Verarbeitungszeiten so gering, daß praktisch keine Behinderungen auftreten. In ein Unterprogramm eingebaut und in regelmäßigen Abständen ausgelöst, reduziert er die Unterbrechungen auf Bruchteile einer Sekunde.

(R. Nitsche)

Startdisk anders erstellt

Die im Benutzerhandbuch des Joyce beschriebene Prozedur zum Erstellen von Startdisketten, die außer der Datei mit dem CP/M-Betriebssystem (**J???CPM3.EMS**, ???=Version) nur noch die zum Arbeiten benötigten Dateien enthält, ist ziemlich umständlich. Möchte man z.B. in Basic programmieren, dann ist es zweckmäßig, außer CP/M nur noch die Datei **BASIC.COM** sowie evtl. ein oder zwei andere benötigte CP/M-Dienstprogramme wie **SHOW.COM** oder **PIP.COM** auf der Diskette zu haben. So ist bei der Programmentwicklung genügend Platz für die einzelnen Module und Versionen eines Programmes vorhanden. Ebenso ist es angenehmer, nach Einschalten des Rechners automatisch in ein Anwendungsprogramm zu gelangen, ohne vorher Disketten zu wechseln und ein oder mehrere Befehle zum Laden und Starten eines Programmes eingeben zu müssen.

Um zuerst einmal genügend freien Speicherplatz auf einer Startdiskette zu bekommen, sollte man laut Handbuch (Kapitel 2, Seite 7) eine vollständige Kopie einer Systemdiskette (Seite 2) mittels **DISCKIT.COM** erstellen – wobei unformatierte Disketten automatisch formatiert werden. Danach sind alle nicht benötigten Programme mit dem **ERASE**-Befehl wieder zu löschen. Dieses Vorgehen erfordert einiges an "Tipperei" und Geduld. Eine Alternative dazu bietet sich durch die von CP/M beim Einschalten des Joyce eingerichtete RAM-Disk an, die mit **m:** angesprochen wird. Man kopiert einfach die benötigten Dateien von der Systemdiskette in die RAM-Disk, legt dann eine leere, formatierte Diskette in das Diskettenlaufwerk und schreibt die Dateien aus der RAM-Disk auf die neue Diskette zurück.

Die notwendigen Schritte zum Erstellen einer wie oben beschriebenen Startdiskette, auf der sich außer CP/M nur noch der Basic-Interpreter befindet, sollen nun zum Verständnis der Leser, die sich durch den Joyce zum erstenmal mit einem Computer bzw. mit dem Betriebssystem CP/M Plus konfrontiert sehen, näher erläutert werden.

Vorweg noch ein kleiner Tip:
Man sollte sich immer einen Vorrat von leeren, formatierten Disketten halten. Dadurch bleibt einem der Umstand erspart, bei der Meldung **Diskette voll** ein Programm abbrechen zu müssen und dabei evtl. wertvolle Daten zu verlieren – nur weil man erst mit DISCKIT durch Formatieren eine Möglichkeit zum Speichern dieser Daten schaffen muß. Außerdem ist es zeitsparender, alle neu gekauften Disketten für den Joyce in einem Arbeitsgang zu formatieren.

Nun zum Erstellen der (Basic-) Startdiskette:
Für das Kopieren von Dateien ist das Dienstprogramm **PIP.COM** (Peripheral Interchange Program) zuständig. Damit lassen sich auf einer Diskette Dateien duplizieren, Dateien von einer Diskette auf eine andere kopieren und noch vieles mehr, was für unser Vorhaben jedoch uninteressant ist.

Um mit dem Kopieren beginnen zu können, wird als erstes dieses Programm durch Eingabe von *pip* in den Rechner geladen, worauf die Meldung **CP/M 3 PIP VERSION 3.0** und in der nächsten Zeile ein Sternchen * erscheint. Dieses Zeichen zeigt die Bereitschaft des Programmes zur Entgegennahme von Befehlen an und das Kopieren kann beginnen. Dazu erwartet **PIP** Befehle in der Form:
Zieldatei=Quelldatei Ä Optionen Ü
(Die Zeichen Ä und Ü entsprechen der eckigen Klammer auf bzw. zu).
Nach Eingabe eines Befehls wird der Kopiervorgang ausgeführt und es können weitere Befehle eingegeben werden, oder PIP durch einfache Betätigung der *RETURN*-Taste verlassen werden.

Ein paar Beispiele:
Mit *datei.bak=datei.txt* wird auf dem angemeldeten Laufwerk unter der gleichen Benutzernummer eine Kopie der Datei *datei.txt* mit dem Namen *datei.bak* angelegt.
datei.txtÄG2Ü=datei.txt legt dagegen eine Kopie der gleichen Datei auf der Diskette unter demselben Namen an, jedoch unter einer anderen Benutzernummer.
Mit *m:=datei.txt* wird die Datei vom angemeldeten Laufwerk unter gleichem Namen auf das Laufwerk *m:* kopiert.
Durch die Verwendung der Platzhalterzeichen * und ? können mit einem Befehl auch mehrere Dateien kopiert werden. Mit *m:=*.txt* werden z.B. alle Dateien des Dateityps *.txt* in die RAM-Disk kopiert.
Mit Hilfe der PIP-Optionen können beim Kopieren die in die Zieldatei übertragenen Daten auf vielfältige Art und Weise manipuliert werden (siehe Benutzerhandbuch, K.5, S. 14). Wichtig für unser Vorhaben ist die Option *O*, mit der PIP die Übertragung einer Objektdatei (Maschinencode, kein Text) mitgeteilt wird. Ansonsten würde PIP die Übertragung der Datei beim Auftreten eines *CTRL-Z*-Zeichens, welches in Text-Dateien das Dateiende anzeigt, beenden. Dieses Zeichen kann in Objektcode- und Basic-Dateien jedoch gültige Daten darstellen.

Für unsere Basic-Startdiskette müssen wir nun die Datei vom Typ *.EMS*, die das CP/M-Betriebssystem enthält, und *BASIC.COM* von der Systemdiskette in das Laufwerk *m:* kopieren:

(1)

m:=.emsÄoÜ*
m:=basic.comÄoÜ

Wenn PIP nach Beendigung der Übertragung von *BASIC.COM* sich wieder mit dem Sternchen meldet, kann die Systemdiskette aus dem Laufwerk genommen und dafür eine leere, formatierte Diskette eingelegt werden. Mit dem Befehl

(2)

a:=m:.*ÄoÜ*

werden dann alle Dateien aus der RAM-Disk *m:* auf die Diskette im Laufwerk *a:*, also die *EMS*-Datei und *BASIC.COM*, geschrieben.

Um das Ergebnis zu überprüfen, wird das PIP-Programm beendet, mit *DIR* der Inhalt der neuen Startdiskette überprüft und, wenn als richtig befunden, durch gleichzeitiges Drücken der *SHIFT*-, *EXTRA*- und *EXIT*-Tasten der Rechner zur Probe mit dieser Diskette neu initialisiert. Daraufhin muß die CP/M-Einschaltmeldung wieder erscheinen und der Rechner kann in Zukunft mit dieser Diskette nach dem Einschalten initialisiert werden. Werden außer Basic noch weitere CP/M-Dienstprogramme auf der neuen Startdiskette benötigt, so brauchen sie bei (1) nur mit in das Laufwerk *m:* kopiert werden.

Mit *PIP* und der RAM-Disk können so auch nachträglich einzelne Dateien von einer Diskette auf eine andere übertragen werden.

Bemerkung:
Eine Alternative zu *PIP* bietet die Disk-Verwaltung von LocoScript. Mit dem Menuepunkt **f3=Kopie** kön-

nen auch CP/M-Dateien bequem nach LocoScript-Manier kopiert werden – Quelle und Ziel werden einfach durch Cursorauswahl bestimmt. Aktiviert man unter **f8=Optionen** die Anzeige der versteckten Dateien, d.h. die der Systemdateien, so kann, ähnlich wie bei der oben beschriebenen Methode, mit **PIP** über die RAM-Disk eine Kopie dieser Dateien auf eine andere Diskette erfolgen.
Damit wäre unser erstes Ziel, eine einfache Startdiskette, erreicht.
Aber wie schon angedeutet, kann man CP/M auch dazu veranlassen, mehrere Befehle automatisch abzuarbeiten. Dabei empfängt das Betriebssystem die Kommandos nicht wie üblich von der Tastatur, sondern liest sie nacheinander aus einer Datei und führt sie aus. Diese Betriebsart wird Stapelverarbeitung (engl. Batch-Processing) genannt und stammt aus der Anfangszeit der EDV, in der die zu verarbeitenden Daten und die entsprechenden Kommandos noch über Lochkartenstapel vom Rechner eingelesen wurden. Diese Betriebsart ist auch im heutigen 'Dialogzeitalter', in dem EDV-Systeme schnell auf unterschiedliche Benutzeranfragen reagieren sollten, sehr nützlich. Mit dieser Möglichkeit können z.B. zeitintensive Verarbeitungen, bei denen große Datenmengen manipuliert werden und nur wenige, immer gleiche zur Steuerung des Ablaufes notwendige Eingaben vom Anwender nötig sind, automatisiert werden.
Aber auch einfachere Dinge, wie im folgenden zu sehen ist, können mit Hilfe der Stapelverarbeitung noch einfacher gemacht werden.
Manche CP/M-Anwendungen benutzen z.B. eine spezielle Tastaturbelegung, die vor Aufruf des Programmes mit dem **SETKEYS**-Kommando aktiviert werden muß. Dadurch ist schon die Eingabe von zwei Befehlen zum Starten eines Programmes notwendig. Wenn auch noch verschiedene andere Voreinstellungen für Drucker (**PAPER, SETLST**) oder serielle Schnittstelle (**SETSIO**) vorgenommen werden müssen, so steigt die vor dem eigentlichen Arbeiten notwendige, immer gleiche "Tipperei" entsprechend. Warum soll das der Rechner nicht selbst machen – und warum nicht gleich nach dem Einschalten und Einlegen einer Programmdiskette?
Um dies zu erreichen, muß erst eine"Stapel"-Datei (engl. Batch-File) erstellt werden. Bei CP/M wird diese durch den Dateityp *.SUB* gekennzeichnet (Beispiel: *START.SUB*), bei anderen Betriebssystemen z.B. mit *.BAT* oder *.EXE*. Dabei handelt es sich um eine einfache Textdatei, in der die vom Betriebssystem auszuführenden Befehle in der gleichen Form enthalten sind, wie sie der Anwender im Dialogbetrieb eingeben würde. Die Erstellung kann mit irgendeinem Text-Editor (ED.COM, RPED .BAS etc.) geschehen. Als einfaches Beispiel soll eine Stapeldatei mit RPED erstellt werden, mit deren Hilfe durch Eingabe eines Kommandos die Bildschirmdarstellung auf 24 mal 80 Zeichen, und der Drucker auf 12 Zoll (72 Zeilen) Endlospapier eingestellt werden. Dazu sind die CP/M-Dienstprogramme **SET 24X80.COM, PAPER.COM** und noch ein anderes Dienstprogramm (mehr dazu etwas später) notwendig, d.h. sie müssen auf der im Laufwerk befindlichen Diskette sein.

```
Bild 1:

submit bdinit

A>paper c,f 72,d
Continuous Stationary
Paper Out Defeat Off
Line Pitch    6 (lines per inch)
Form Length 72 (lines)
Gap Length    0 (lines)
Defaults set

A>set24x80
```

```
Bild 2: Datei logo.sub

setkeys keys.drl
logo.com
```

Nach Laden von Basic und Starten des Editors wählen wir im Bearbeitungsmenue den Punkt **neuen Bildschirminhalt bearbeiten** und geben als Dateinamen *BDINIT.SUB* (**B**ild und **D**rucker **in**itialisieren) an. Im darauf erscheinenden 'leeren' Bildschirm werden nun die zwei dazu nötigen Kommandos eingegeben:

paper c,f 72,d
set24x80

Nach Beenden von RPED steht nun die Stapeldatei auf der Diskette und kann abgearbeitet werden. Dazu ist das oben schon angedeutete, weitere Dienstprogramm **SUBMIT.COM** notwendig. Durch dieses Programm wird die von uns erstellte Datei für die Bearbeitung von CP/M noch einmal überarbeitet und in eine temporäre Datei abgelegt. Dies ist u.a. notwendig, da eine Stapeldatei auch mit Übergabe von Parametern aufgerufen werden kann und die Platzhalter in der Stapeldatei durch die aktuellen Parameter des Aufrufes ersetzt werden. Danach wird dann Befehl für Befehl dem Betriebssystem zur Ausführung eingereicht – daher der Name **SUBMIT** (zu Deutsch: vorlegen, einreichen) und der Dateityp **.SUB**. Um das Beispiel jetzt abarbeiten zu lassen, wird

submit bdinit

eingegeben.

Auf dem Bildschirm kann nun die Abarbeitung der Stapeldatei anhand der Ausgaben des Betriebssystems verfolgt werden und wie in Bild 1 zu ersehen ist, entspricht die Anzeige exakt der, die durch Eingabe der einzelnen Kommandos auf dem Bildschirm entstehen würde.
Ein weiteres, einfaches Beispiel einer Stapeldatei befindet sich auf der zweiten Systemdiskette (Seite 4) zum Starten der Programmiersprache LOGO (Bild 2). Bei Eingabe von **submit logo** wird hier die Tastaturbelegung **setkeys keys.drl** modifiziert, bevor durch **logo.com** die Sprache aufgerufen wird.
Somit wissen wir jetzt, wie wir immer gleichbleibende Kommandofolgen durch Eingabe eines einzelnen Befehls abarbeiten lassen können – was noch fehlt, ist das automatische Starten eines Programmes mit allen nötigen 'Vorbereitungen' (SETirgendwas etc.) nach Einschalten des Rechners und Einlegen der Programmdiskette. Überhaupt kein Problem! Dazu wird nur eine Stapeldatei namens *PROFILE.SUB* auf der Diskette benötigt.

Nach Laden und Initialisieren von CP/M überprüft das Betriebssystem nämlich auf der Diskette das Vorhandensein einer Stapeldatei mit diesem Namen und führt diese dann aus. Auch dazu befindet sich auf der CP/M-Systemdiskette (Seite 2) schon ein Beispiel – die Datei **PROFILE.GER** (Bild 3). Die Ausführung dieser Datei bewirkt das Kopieren der meistgebrauchten Dienstprogramme in die RAM-Disk (Laufwerk m:), um sie auch nach Diskettenwechsel zur Verfügung zu haben. Außerdem wird mit dem **SETDEF**-Dienstprogramm die Reihenfolge der Laufwerke, nach der CP/M diese bei der Suche nach einem Programm 'abklappert' und des dabei zu berücksichtigende Dateityps, festgelegt (s. Handbuch K. 5, S. 22). Um diese Stapeldatei beim 'booten' ausführen zu lassen, muß sie in **PROFILE.SUB** umbenannt werden:

rename profile.sub=profile.ger

```
Bild 3: Datei profile.ger

setdef m:,* Äorder = (sub,com) temporary = m:ü
pip
<m:=basic.comÄoÜ
<m:=dir.comÄoÜ
<m:=erase.comÄoÜ
<m:=paper.comÄoÜ
<m:=pip.comÄoÜ
<m:=rename.comÄoÜ
<m:=setkeys.comÄoÜ
<m:=show.comÄoÜ
<m:=submit.comÄoÜ
<m:=type.comÄoÜ
<
```

```
Bild 5: Datei rped.sub

basic rped
```

```
Bild 4: Ausführung von profile.sub

A>setdef m:,* Äorder = (sub,com) temporary = m:ü

Drive Search Path:
1st Drive             - M:
2nd Drive             - Default

Search Order          - SUB, COM
Temporary Drive       - M:

A>pip
CP/M 3 PIP VERSION 3.0
*m:=basic.comÄoÜ
*m:=dir.comÄoÜ
*m:=erase.comÄoÜ
*m:=paper.comÄoÜ
*m:=pip.comÄoÜ
*m:=rename.comÄoÜ
*m:=setkeys.comÄoÜ
*m:=show.comÄoÜ
*m:=submit.comÄoÜ
*m:=type.comÄoÜ
*
```

Wenn jetzt der Rechner durch Drücken der **SHIFT-**, **EXTRA-** und **EXIT-**Taste neu initialisiert wird, wird diese Stapeldatei automatisch ausgeführt (Bild 4), bevor CP/M den ersten Befehl des Benutzers erwartet. Soll dagegen gleich ein Programm gestartet werden, so braucht in der **PROFILE.SUB-**Stapeldatei nur der dazu notwendige Befehl zu stehen. **BASIC** würde z.B. das sofortige Laden des Mallard-Basic-Interpreters bewirken, wogegen der Befehl **BASIC meinprog.bas** zusätzlich das Basic-Programm **meinprog.bas** laden und starten würde. Ein Beispiel dazu ist die Datei **RPED.SUB** auf der Systemdiskette (Bild 5). Wurde die oben genannte Datei **PROFILE.SUB** (ehem. PROFILE.GER) ausgeführt, so genügt zum Starten des Basic-Programmes **RPED.BAS** nur noch die Eingabe *rped* – das Laden und Starten des Basic-Interpreters übernimmt der Rechner selbständig.
Ein weiteres Beispiel wäre eine LOGO-Startdiskette:
Kopiert man sich die zu **Dr. LOGO** gehörenden Dateien **LOGO.SUB**, **LOGO.COM** und **KEYS.DRL** auf eine Diskette zusammen mit den Dateien **J???CPM3.EMS**, **SUBMIT.COM** und **SETKEYS.COM** und nennt die Datei **LOGO.SUB** in **PROFILE.SUB** um, so wird die Programmiersprache Logo nach Einschalten des Rechners gestartet.
Diese Beispiele sollen als Anregung genügen und der Erstellung einer Multiplan-, dBase- oder Wordstar-Startdiskette dürfte kein Hindernis mehr im Wege stehen.
Zusammenfassend hier noch einmal die auf einer Startdiskette nötigen Dateien:

Unbedingt notwendig sind **J???CPM3 .EMS** (das Betriebssystem), **SUBMIT .COM** (zur 'Einreichung' der Befehle aus der Stapeldatei an das Betriebssystem) und die Stapeldatei **PROFILE.SUB** mit den auszuführenden Kommandos.

Zusätzlich notwendig sind außerdem noch alle die aus der Stapeldatei aufgerufenen Dienst- und Anwendungsprogramme sowie die dazugehörigen Datendateien.

(MC)

Wie der MacIntosh und der Joyce miteinander reden

Der Schneider Joyce ist als Textverarbeitungssystem konzipiert, arbeitet unter dem Betriebssystem CP/M und weist ein ausgezeichnetes Preis-/Leistungsverhältnis auf. Dagegen ist der MacIntosh sowohl in seinen Preisen als auch in seiner Leistung deutlich höher angesiedelt. Aus gegebenem Anlaß entstand der Wunsch, umfangreiche Texte, die bisher auf einem MacIntosh erstellt wurden, auf den Joyce zu übertragen, und dort weiter zu verarbeiten. Die Kopplung der zwei Rechner, die Datenübertragung und einige vergleichende Beobachtungen sind Gegenstand des folgenden Berichts.

Um Daten von einem Computer auf einen anderen zu übertragen, braucht man zunächst einmal ein Kabel sowie geeignete Software. Da die Pinbelegungen des MacIntosh und der Joyce-Schnittstelle dokumentiert sind, stellt das kein allzu großes Problem dar. Beide Computer verfügen über eine RS232-Schnittstelle. Für den MacIntosh benötigt man einen neunpoligen (male)-Stecker, für den Joyce einen 25-poligen (female)-Stecker und zusätzlich ein dreiadriges Kabel. Das alles ist in einem Elektronik-Fachgeschäft für etwa DM 15,– zu erhalten.

MacIntosh	Joyce
Pin 8+1 miteinander	
5 TxD- (Sender)	3 RxD (Empfänger)
9 RxD- (Empfänger)	2 TxD (Sender)
3 Ground (Masse)	7 Ground (Masse)

Auf der MacIntosh-Seite sollten Pin 8 und Pin 1 miteinander verbunden werden, damit die Schnittstelle als RS232 agiert.
Freundlicherweise ist in der mitgelieferten Software des Joyce bereits ein einigermaßen komfortables Terminalprogramm enthalten. Nach dem Starten des CP/M-Betriebssystems muß die Diskette gewendet werden, damit "MAIL232" aufgerufen werden kann. Nach Drücken der Taste F1 können dann die Übertragungsparameter eingestellt werden. Man kann bei kleineren Files bedenkenlos mit 9600 Baud senden und empfangen. Um reinen Text (also ASCII-Files) zu übertragen, sollten sieben Bits und gerade Parität genügen. Nach Drücken der Taste F3 kann dann ein Name für das zu sendende/übertragende File eingesetzt werden. Durch Enter wird die Übertragung gestartet und mit Alt/Stop beendet.

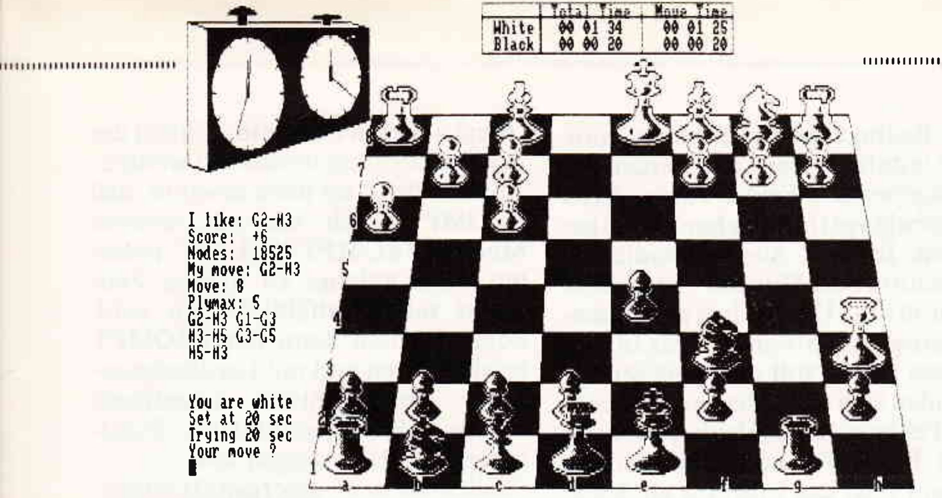
Auf dem MacIntosh muß zunächst einmal ein Terminalprogramm gestartet werden. Das ist hier das Public Domain-Programm Red Ryder. Public Domain-Programm bedeutet, daß das Programm bei Benutzergruppen des MacIntosh oder auch über Mailboxen kostenlos erhältlich ist. Erst nach ausgiebigen Tests kann man sich entscheiden, das Programm zu behalten und dem Autor freiwillig einen kleinen Betrag zu überweisen oder es ganz zu löschen. In seiner Leistungsfähigkeit übertrifft es übrigens das von Apple vertriebene Programm Mac Terminal deutlich. Die einzustellenden Parameter sind die gleichen wie beim Joyce: 9600 Baud, gerade Parität, sieben Bits, ein Stoppbit; zusätzlich kann noch CR after LF (nach jeder Zeile wird automatisch ein Zeilenvorschub erzeugt) eingestellt werden. Mit diesen Einstellungen können beide Rechner problemlos ASCII-Files senden und empfangen.
Häufig liegt aber der Teufel im Detail. Daher noch einige weitere Tips. Auf dem MacIntosh werden Umlaute als ASCII-Zeichen oberhalb von 7F dargestellt und entsprechen nicht mehr den (allgemein üblichen) Verabredungen. Zeichen oberhalb von 7F würden bei der gewählten Einstellung der Übertragungsparameter folglich nicht richtig übertragen. Es ist deshalb angebracht, die Umlaute und Sonderzeichen (scharfes s (ß), Mikrometer-Symbol (μ) etc.) in übertragbare Zeichen zu übersetzen. Am einfachsten wäre es natürlich, ein ö durch oe zu ersetzen. Beim Rücktausch müßten dann alle oe's in ö's verwandelt werden. Aber Vorsicht: aus Goethe wird anschließend Göthe! Besser ist es, ö durch z.B. o# zu ersetzen. Die Zuordnung rückwärts ist nun eindeutig und zweifelsfrei möglich, so daß automatisch über den gesamten Text getauscht werden kann. Selbstverständlich dürfen im Originaltext nicht Zeichenfolgen wie o# auftreten.

Mein Vorschlag:
ä durch a#, Ä durch A#
ö durch o#, Ö durch O#
ü durch u#, Ü durch U#
ß durch s#, μ durch m# ersetzen.

Nach Umwandlung des Quelltextes in ein reines ASCII-File ohne Formatierungsangaben, kann man die Änderungen mit der Suche/Tausche-Funktion eines Texteditors rasch erledigen. Ein File mit etwas mehr als 100 KByte ist auf der MacIntosh-Seite in knapp zehn Minuten präpariert. Die Rückwandlung auf dem Joyce unter LocoScript dauert allerdings mehrere Stunden! An diesem Punkt macht es sich doch bemerkbar, daß der MacIntosh über einen modernen 16-Bit-Prozessor verfügt.
Anfänglich hatte ich allergrößte Probleme damit, empfangene Files auf den Joyce auch abspeichern zu lassen. Obwohl genügend Platz auf der Diskette war, hat sich der Joyce in neun von zehn Fällen kommentarlos geweigert, das File ordnungsgemäß abzulegen. Aus gesendeten 10 KByte wurden so meist 0 KByte, manchmal auch 3 oder 5 KByte. Im zehnten Anlauf wurde das zuvor mehrfach erfolglos gesendete File endlich auf der Diskette gesichert. Mein Tip dazu: eine Joyce CF2-Diskette initialisieren, MAIL232 starten und danach die initialisierte, leere Diskette in das Laufwerk A: einlegen. Leider kann man von MAIL232 aus nicht auf das zweite Laufwerk B: zugreifen. Mit dem oben genannten Trick hat der Joyce auch ein File von mehr als 100 KByte Größe problemlos empfangen und gespeichert. Möglicherweise "stören" auf den Joyce-Disketten die Trans-Files; wobei mir die Ursachen für das Fehlverhalten unbekannt sind. Da der Joyce nach einer Weile die empfangenen Daten aus dem Puffer auf die Diskette schiebt, können während der Zeit des Diskettenzugriffs einige der gerade empfangenen Daten verlorengehen. Meist kann man dem dadurch begegnen, indem man die Baudrate auf 1200 oder gar 300 Baud absenkt, wobei natürlich die Übertragungszeiten steigen. Die Übertragung und Speicherung von Dateien beliebiger Größe zum MacIntosh war dagegen niemals mit Problemen behaftet.
Etwas ärgerlich ist, daß das MAIL232-Programm auf dem Joyce nicht auf der CP/M-Seite der Diskette steht. Der Versuch, unter CP/M mittels PIP das Programm auf die andere Diskettenseite oder eine andere Diskette zu übertragen, scheiterte, weil ausgerechnet (und nicht ganz einsichtig) MAIL 232 als System-File gekennzeichnet ist. (Anm. Red.: System-Files lassen sich mit 'pip zielname=quellnameÄrÜ' kopieren, die Kennzeichnung als System-File bleibt dabei erhalten).
Auch dagegen ist ein Kraut gewachsen! Wenn man mit LocoScript startet, dann lasse man sich die versteckten Files (mit F8) zeigen. Unter anderem wird auch MAIL232.COM aufgelistet, allerdings mit einem V-Attribut (versteckt), was wohl für System-Files stehen soll. Wählen Sie einfach MAIL 232 an, übertragen es in die RAM-Disk (Laufwerk M:) und benennen es um (mit F5). Jetzt ist das Attribut V weg! Von der RAM-Disk läßt sich das Programm (am besten wieder zurückbenannt) auf jeder anderen Diskette speichern. Das geht mit einem derart entsicherten Programm jetzt sogar von CP/M aus. Normalerweise würde man das Systemattribut unter CP/M 2.2 mit dem Programm STAT ändern. Die vielfältigen Funktionen dieses Programms sind bei CP/M Plus aber auf verschiedene Dienstprogramme verteilt worden, deren Funktionen sich für den Benutzer durch entsprechenden Namen des Dienstprogrammes leichter erlernen (merken) lassen. Unter CP/M Plus lassen sich mit dem Programm SET.COM die Attribute der Dateien ändern, wobei erst der etwas erfahrenere Benutzer nach Studium des Handbuchs auf die entsprechende Funktion stößt (z.B. 'set mail232.comÄdirÜ'). Mit etwas "Rumspielen" (Hinweis: PROFILE.GER ändern und anschließend in PROFILE .COM umbenennen) kann man sogar die CP/M-Diskette einlegen und automatisch mit MAIL232 starten.
Mit dem gleichen Trick lassen sich auch die Files J21CLOCO.EMS, MATRIX .STD und SCRIPT.JOY in die RAM-Disk übertragen, und von dort auf einer anderen Diskette ablegen. Das File MATRIX.STD muß unter dem Trio sein, es ist wahrscheinlich für die Druckersteuerung zuständig. Mit diesen drei Files kann man sich eine LocoScript-Startdiskette erstellen, ohne zuvor umständlich die ganze Diskette kopieren zu müssen.

Einige abschließende Bemerkungen zum Vergleich MacIntosh – Joyce: Der MacIntosh bietet zweifelsfrei mehr Leistung, deutlich mehr Komfort für (allerdings!) viel mehr Geld. Ich darf gar nicht daran denken, daß die Speicheraufrüstung auf 512 KByte beim MacIntosh einmal fast DM 4000,– gekostet hat! Was mir am Joyce gut gefällt ist, daß er nach dem Einschalten deutlich schneller als der MacIntosh zur Verfügung steht. Dem Betriebssystem CP/M sind mit der Diskettenverwaltung unter LocoScript auch einige Schrecken genommen (ich denke an die allfälligen Tippfehler, auf die CP/M gnadenlos reagiert). Mit dem großen Laufwerk B: habe ich z.Z. sogar auf dem Joyce mehr Diskettenkapazität als auf dem MacIntosh. Wenn ich die Druckqualitäten miteinander vergleiche, ist der Joyce eindeutiger Sieger und der MacIntosh mit dem Imagewriter weit, weit abgeschlagen. Erst wenn ich mir einen Laserdrucker im Gegenwert von zehn Joyces leisten würde, könnte der MacIntosh wieder davonziehen.

(Dr. Martin Wolter/MC)



3D-Clock Chess

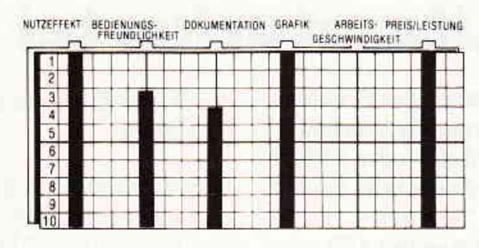
Gebiet: Brettspiele
Hersteller: CP Software
Vertrieb: Fachhandel
Rechner: Joyce 8256/8512
Sprache: M-Code
Preis DM 69,90

Mit 3D-Clock Chess präsentiert sich das erste Spiel für den Joyce überhaupt. Das Spiel der Könige wurde hier in einer eindrucksvollen Art und Weise auf die Joyce-Computer angepaßt und bringt etwas Abwechslung in den sonst mit Anwenderprogrammen übersäten Joyce-Softwaremarkt. Clock Chess läuft unter CP/M Plus und weist eine hervorragende 3D-Grafik auf. Erstmals wurde hier die gesamte Bildschirmbreite des Joyce ausgenutzt, das Ergebnis ist ein übersichtliches Spielfeld mit sehr gut zu erkennenden Figuren sowie einer grafisch dargestellten Schachuhr (siehe auch Hardcopy).

Als wir das Programm zum ersten Mal geladen hatten, staunten wir über die perfekte Grafik nicht schlecht. Doch als eingefleischter Hobby-Schachspieler stellt man sich zunächst die Frage, wie gut das Programm trotz aller Grafikeigenschaften überhaupt spielt. Und das war eine faustdicke Überraschung! 3D-Clock Chess ist eines der stärksten Schachprogramme, die wir jemals auf einem Computer zu sehen bekamen. Daneben stellt das Programm jede Menge Optionen zur Verfügung, mit denen das Spiel variiert werden kann. So ist die Spielstärke voll flexibel, vom Spiel auf Zeit bis hin zu Problem- und Blitzschach, 3D-Clock Chess schöpft aus einem vollen Repertoire. Sehenswert ist allein schon schon die Display-Anzeige während einer Schachpartie; der Spieler kann hier zu jeder Zeit wichtige Informationen über Spielstand, Anzahl der Züge, Zeit sowie mögliche Züge des Computers ablesen.

Über entsprechende Untermenues besteht zusätzlich die Möglichkeit, weitere Informationen wie z.B. Zugvorschläge des Computers, Ändern der Spielstärke, Invertieren etc. abzurufen; natürlich kann das Spiel auch auf Diskette abgespeichert und wieder geladen werden.

Insgesamt macht 3D-Clock Chess einen hervorragenden Eindruck. Sowohl grafisch als auch von der Spielstärke sucht es seinesgleichen. Die Spielstärke wird ausnahmslos über die Zeit eingestellt, 5 - 999 Sekunden sind wählbar. Fast alle erdenklichen Optionen eines guten Schachprogrammes sind vorhanden. Einziges Manko, das wir feststellen konnten, ist das Fehlen einer Zugrücknahme. Doch das gibt es im Live-Schach sowieso nicht und ist daher auch nicht weiter schlimm.



Prompt

Gebiet: Datenverwaltung
Hersteller: DDI-Software
Vertrieb: Fachhandel/Schneider
Rechner: JOYCE
Sprache: Basic
Preis: 69,- DM

Oft steht der Computeranwender, gerade wenn er sich erst seit kurzer Zeit mit diesem technischen Hilfsmittel befaßt, vor dem Problem, eine seinen Ansprüchen und Fähigkeiten entsprechende Software zu finden. Es gibt Programme, mit denen eine Vielzahl von Problemlösungen möglich sind, jedoch erfordern sie für eine effektive Arbeit doch etwas fundiertere Kenntnisse der EDV.

Es geht auch anders

Schon beim Entnehmen von "Prompt" aus der Verpackung fällt das dünne, nur 34 Seiten umfassende "Handbuch" des Programmes auf. In der Einleitung wird dann auch die Philosophie dieser Software deutlich, die sich in Programm und Dokumentation wiederspiegelt: dem Anwender mit wenig Wissen um Programmierung ein einfaches und dennoch leistungsfähiges Werkzeug zur Erstellung seiner individuellen Datenverwaltung zu geben. Trotz des relativ geringen Umfanges des Handbuches werden alle notwendigen Maßnahmen zum Betrieb des Programmes und die Möglichkeiten von PROMPT gut und sehr verständlich dargelegt, so daß auch Computerneulinge keine großen Probleme mit dem Programm haben dürften. Befolgt man schrittweise die Anweisungen der Dokumentation, so kann nach wenigen Minuten schon das erste selbstgeschriebene Datenverwaltungsprogramm erstellt worden sein.

Dazu stellt PROMPT einen Masken- und Programmkern-Generator zur Verfügung, mit dem die zu verwaltenden Daten und ihre Beziehungen zueinander festgelegt werden. Dazu wird als erstes auf dem Bildschirm, mit Hilfe des Maskengenerators, ein Formular mit den notwendigen Datenfeldern entworfen, in denen später bei der Erfassung, Verwaltung und Änderung der Daten die Eingaben vom Anwender getätigt werden. Ein Datenfeld besteht aus einer Nummer, über die das Feld bei der weiteren Bearbeitung selektiert werden kann, einer Feldbezeichnung sowie dem eigentlichen Ein-/Ausgabefeld, wozu zwei Feldtypen zur Auswahl stehen: Zeichen- und Rechenfelder. Markiert werden sie jeweils durch eine Unterstreichungs-Linie bzw. durch Punkte, die auch die Länge der Felder festlegen.

Ist man mit dem entworfenen Formular zufrieden, so tritt der Programmkerngenerator in Aktion. Hier wird das "Suchfeld" definiert, über das später nach den einzelnen Datensätzen gesucht werden kann. Ebenso wird die Anzahl der Nachkommastellen der Rechenfelder sowie ihre Verknüpfung/Berechnung festgelegt. Dazu stehen die vier Grundrechenarten und Klammerebenen zur Verfügung. In der Formel zur Berechnung eines Feldes können Inhalte anderer Rechenfelder durch ihre Feldnummer (z.B.: F5) aufgenommen werden. So könnte sich z.B. ein Feld "Brutto-Umsatz:" (F10) aus den Feldern (F7 bis F9) mit Netto-Umsätzen, die nach einzelnen Kategorien aufgeteilt sind, aus F10=(F7+F8+F9)*1.14 berechnen und ein Gesamtumsatz in Feld 11 (über einen größeren Zeitraum) mit F11=F11+F10.

Schließlich teilt man PROMPT noch mit, welche Felder Bewegungsdaten

enthalten, d.h. immer wieder neue Daten eingegeben werden, wie Verkaufszahlen eines Monats - im Gegensatz zu Stammdaten, wie Adressen, die sich selten ändern und das Programm zur Verwaltung von Umsätzen z.B. fertig ist.
Mit den bei jeder Art von Datenverwaltung notwendigen Funktionen zum Erfassen, Verwalten und Ändern können nun die eigentlichen Daten eingegeben werden. Im Verwaltungsmodus läßt das Programm nur Eingaben in den als Bewegungsdaten definierten Feldern zu, so daß ungewollte Änderungen der anderen Felder nicht möglich sind. Bei allen Funktionen werden die durch Formeln verknüpften Rechenfelder neu berechnet. Dabei ist es auch möglich, statt einfacher Zahlen, wie im Programmkerngenerator, Formeln in ein Rechenfeld einzugeben, was den Taschenrechner ein wenig entlastet.
Oft benötigt man Auszüge aus einem Datenbestand in einer übersichtlichen Form von Tabellen bzw. Listen. Dazu stellt PROMPT einen Listen-Generator zur Verfügung, mit dem man die relevanten Daten anhand von bis zu zehn Bedingungen selektieren kann. Zur Gestaltung der Liste werden die auszugebenden Felder sowie deren Ausgabelänge (bei Zeichenfelder) bestimmt. Je nach Ausgabemedium - Bildschirm oder Drucker - können so bis zu 90 bzw. 136 Zeichen pro Datensatz ausgegeben werden, was in den meisten Fällen voll und ganz genügt. Befinden sich unter den auszugebenden Feldern Rechenfelder, so summiert PROMPT automatisch diese auf und zeigt das Ergebnis am Ende der Liste. Durch Angabe der entsprechenden Suchbedingungen können komplette Listen des gesamten Datenbestandes oder Auszüge erstellt werden, wobei als Vergleichsoperatoren <, >, <= und >= dienen, und, wie schon erwähnt, auf bis zu zehn Felder eines Datensatzes angewandt werden können.
Damit wären auch schon alle zur Datenverwaltung nötigen Funktionen des Programmes vorgestellt - bis auf eine: Wenn durch die Verwaltung der Daten eine zu große Unordnung im Bestand entstanden ist, so kann mit Hilfe des Sortier-Programmes die Datei nach dem Suchschlüssel oder einem anderen Feld (hinsichtlich der Listenerstellung) umsortiert werden.
Abschließend sei noch erwähnt, daß PROMPT noch um ein weiteres Modul "PROMPT-DRUCK" erweitert wird, welches zu diesem Zeitpunkt schon erhältlich sein wird. Somit können dann mit PROMPT erfaßte Daten und mit LocoScript erstellte Texte gemischt ausgedruckt werden, z.B. Adressaufkleber, Rundschreiben, Rechnungen usw.
Schließlich noch eine (ganz?) subjektive Bewertung von PROMPT: Hinsichtlich des Preises, der einfachen und leicht zu erlernenden Bedienung und den gebotenen Möglichkeiten stellt das Programm, gerade für den Computerneuling, eine empfehlenswerte Alternative zu teureren, komplexeren und relativ schwer zu bedienenden Programmen dar. (MC)

NUTZEFFEKT BEDIENUNGS-FREUNDLICHKEIT DOKUMENTATION GRAFIK ARBEITS-GESCHWINDIGKEIT PREIS/LEISTUNG
1 2 3 4 5 6 7 8 9 10

LITBOX - eine Literaturkartei für den JOYCE

Schüler, Lehrer, Studenten und andere kennen das Problem: Da gab es doch zu diesem Thema, das man gerade bearbeitet, einen hervorragenden Aufsatz von dem Meier oder Müller (wie hieß der doch bloß?) in dieser XY-Zeitschrift (in der Januarausgabe!?). Im Titel kam irgendwo "Umwelt-" vor... - aber zum Glück hat man sich dazu ja irgendwo (!?) eine Notiz gemacht...

Helfen könnte in diesem Falle das Programm LITBOX (Listing 1), sofern man seinen elektronischen Karteikasten erst einmal reichlich gefüllt hat.
Weil dieses Programm den guten alten Karteikasten elektronisch nachahmt, soll im folgenden auch jeweils von "Kartei" die Rede sein, wenn eine Computer-"Datei" gemeint ist, und von 'Karteikarte', wenn es sich genaugenommen um den "Satz" einer "Random-Datei" (Direktzugriffsdatei) handelt. Entsprechend ist die "Reg.-Nr." einer "Karteikarte" die "Satznummer" in dieser "Datei"!

Programmbeschreibung:
Das Programm LITBOX ermöglicht es, mit dem Joyce Literaturkarteien zu erstellen, zu verändern und nach beliebigen Stichwörtern zu durchsuchen.
Eine Haupt-Kartei (NAME.KRT) umfaßt bis zu 600 Karteikarten, auf denen die eingegebenen Texte gespeichert werden. Dazu gehört jeweils eine Kartei (NAME.STI), die 108 Karteikarten mit den alphabetisch geordneten Stichwörtern der Haupt-Kartei enthält. Die zwei zusammengehörigen Karteien benötigen 78K (76K+2K) Speicherplatz. Dadurch ist es möglich, Mallard-Basic, dazu einige CP/M Dienstprogramme und jeweils eine KARTEI (76+2k) in die RAM-Disk (M:) zu laden. Das empfiehlt sich immer dann, wenn eine Kartei ausgiebig durchsucht oder ergänzt werden soll, da vor allem das Durchsuchen einer Kartei in der RAM-Disk etwa doppelt so schnell geht wie mit dem Laufwerk.

Die Handhabung des Programms ist denkbar einfach, da es nach dem Laden und Starten vollständig dialog- bzw. menuegesteuert abläuft, wobei sämtliche Tastenfunktionen auf dem Bildschirm erläutert werden. Falsche Eingaben (z.B. ein falscher Karteiname) werden ignoriert und können leicht korrigiert werden.

Fehler bei der Eingabe von Texten können ebenfalls problemlos berichtigt werden. Außerdem kann man immer dann, wenn das Programm eine Eingabe erwartet, mit Hilfe der CAN-Taste aus dem aktuellen Programmteil "aussteigen" und zum vorangehenden zurückkehren, oder mit der EXIT-Taste das Programm beenden. Dabei sorgt das Programm automatisch für die Sicherung aller eventuell eingegebenen Daten auf dem angegebenen Laufwerk.
Nach dem Laden und Starten (wie man diesen Ablauf unter CP/M-Steuerung einigermaßen komfortabel gestalten kann, soll später erläutert werden) meldet sich das Programm mit einigen Hinweisen zur Programmsteuerung (Z. 240 - 320). Danach muß angegeben werden, ob man im Drucker Einzelblätter oder Endlospapier verwendet. Gleichzeitig wird der Drucker auf Kompaktschrift eingestellt. Anschließend werden die RAM-Disk und das Standardlaufwerk (A:) nach geeigneten Karteien durchsucht. Das Ergebnis dieser Suche wird im Menue 1 angezeigt (Z. 520 - 560).

Jetzt kann man entweder eine neue Diskette einlegen (und die Suche wiederholen) oder zwischen der Bearbeitung einer vorhandenen oder der Einrichtung einer neuen Kartei wählen.

Hat man das Unterprogramm "Neue KARTEI erstellen" aufgerufen, muß man zunächst dem Programm mitteilen, welchen Namen die Kartei haben soll. Der Karteiname kann max. acht Zeichen lang sein, wobei das erste Zeichen unbedingt eine 1 bzw. 2 sein muß, um dem Suchprogramm die Identifikation der Karteien zu ermöglichen (auf eine

Listing 1:
Literaturkartei LITBOX.BAS

```
10 '============================= Initialisierun
g ======================================
20 '------- Einrichtung von Variablen zur Bildschi
rmlöschung und Cursorsteuerung ---------
30 ' Gesamtlöschung; Zeilenlöschung; löschen ab Cu
rsor; löschen bis Cursor:
40 cl$=CHR$(27)+"E"+CHR$(27)+"H":cl.z$=CHR$(27)+"l
":cl.u$=CHR$(27)+"J":cl.o$=CHR$(27)+"d"
50 ' Cursor an; Cursor aus; Cursor links:
60 c.aus$=CHR$(27)+"f":c.an$=CHR$(27)+"e":c.li$=CH
R$(8)
70 '------------ Definition der Positionierungsfun
ktion für Bildschirmausgabe ------------
80 DEF FNpkt$(zeile,spalte,zchn$)=CHR$(27)+"Y"+CHR
$(32+zeile)+CHR$(32+spalte)+zchn$
90 '------------------------ sonstige Initialisier
ungen ----------------------------------
100 inv.an$=CHR$(27)+"p":inv.aus$=CHR$(27)+"q"'---
--> Inverse Schrift an-/ausstellen
110 DIM karte$(10):DIM sw$(110):OPTION RUN
120 '============================ Startmaske ==
======================================
130 PRINT cl$;inv.an$;c.aus$;
140 PRINT FNpkt$(2,13,CHR$(134)+STRING$(57,138)+CH
R$(140))'---> Rahmen/oberer Rand
150 FOR x=3 TO 27
160 PRINT FNpkt$(x,13,CHR$(133));FNpkt$(x,71,CHR$(
133))'------> Seitenränder
170 NEXT
180 PRINT FNpkt$(11,13,CHR$(135)+STRING$(57,138)+C
HR$(141))'--> unterer Rand
190 PRINT FNpkt$(28,13,CHR$(131)+STRING$(57,138)+C
HR$(137))
200 PRINT FNpkt$(4,28,SPACE$(31))
210 PRINT FNpkt$(5,28," L I T E R A T U R K A R T
E I ")
220 PRINT FNpkt$(6,28,SPACE$(31))
230 PRINT FNpkt$(9,33,inv.aus$+"(c) 1986 Klaus Wil
tsch")
240 '-------------------Informationen zur Program
msteuerung -----------------------------
250 PRINT FNpkt$(14,16,">>> Während jeder Eingabe
kann das Programm mit der")
260 PRINT FNpkt$(15,20,inv.an$+" CAN "+inv.aus$+"
-Taste im Programmablauf zurückgesetzt und")
270 PRINT FNpkt$(16,20,"mit der "+inv.an$+" EXIT "
+inv.aus$+"-Taste beendet werden")
280 PRINT FNpkt$(18,16,">>> Mit der "+inv.an$+" "+
CHR$(27)+CHR$(11)+"DEL "+inv.aus$+"-Taste kann Tex
t schrittweise ge-")
290 PRINT FNpkt$(19,20,"löscht werden")
300 PRINT FNpkt$(21,16,">>> Nach jeder Texteingabe
 "+inv.an$+" RETURN od. ENTER "+inv.aus$+" drücken
"+inv.an$);
310 PRINT FNpkt$(24,23," Bitte eine geeignete Disk
ette einlegen ")
320 PRINT FNpkt$(26,24," Zum Programmstart eine Ta
ste drücken ")
330 taste$=INKEY$:IF taste$="" THEN 330
340 IF taste$=CHR$(8) OR taste$=CHR$(27) THEN 3640
'-----> Programm beenden
350 '---------------------------- Druckereinstell
ung -----------------------------------
360 PRINT cl$;FNpkt$(12,27,inv.an$+" Welche Papier
art wird verwendet? ")
370 PRINT FNpkt$(14,35,inv.an$+" 1 "+inv.aus$+" Ei
nzelblatt")
380 PRINT FNpkt$(16,35,inv.an$+" 2 "+inv.aus$+" En
dlospapier ")
390 PRINT FNpkt$(18,24,inv.an$+" Bitte die entspre
chende Taste drücken "+inv.aus$)
400 taste$=INKEY$:IF taste$="" THEN 400
410 IF taste$=CHR$(8) OR taste$=CHR$(27) THEN 3640
'-----> Programm beenden
420 IF taste$="1" OR taste$="2" THEN 430 ELSE 400
430 IF taste$="1" THEN LPRINT CHR$(27);"$";
```

Diskette passen zwei Karteien). Das Programm überprüft dann, ob der eingegebene Name zulässig und auf der im Laufwerk befindlichen Diskette noch nicht vorhanden ist. Falls der Name korrekt ist, hat man die Möglichkeit, einen 83 Zeichen langen Text zur Identifikation der Kartei (Inhaltsangabe) einzugeben. Nachdem die neue Kartei eingerichtet worden ist, kehrt das Programm zum Menue 1 zurück. Wählt man das Programm "Vorhandene Kartei bearbeiten", muß man dem Programm zunächst mitteilen, welche Kartei bearbeitet werden soll, und dann die Unterprogramme "Suchen" oder "Ergänzen" aufrufen. Befindet sich die angegebene Kartei sowohl in der RAM-Disk als auch im Laufwerk A:, wird die in der RAM-Disk zur Bearbeitung geöffnet.

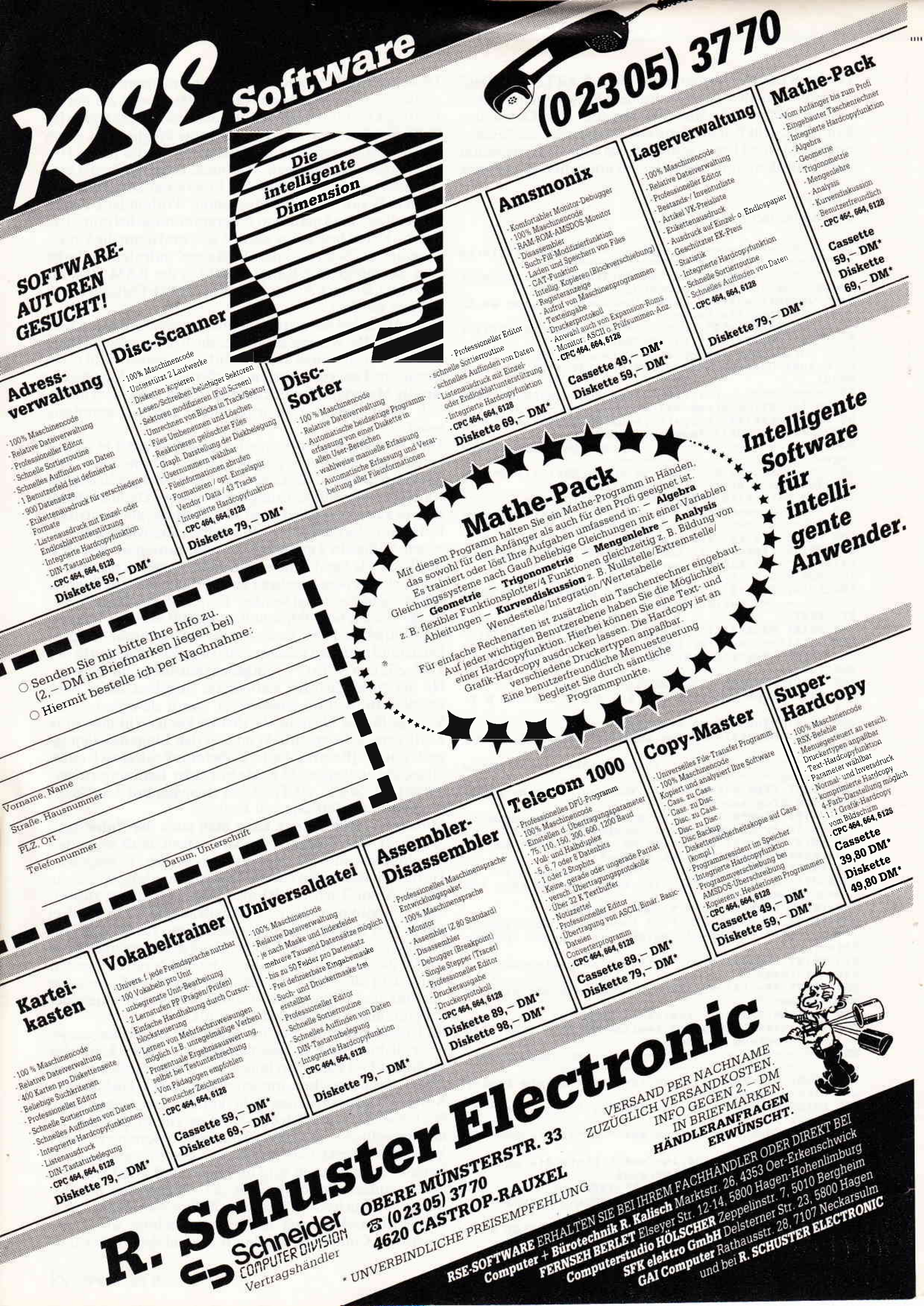
Beim Ausfüllen der Karteikarten kann man den Autor und den Titel des Werkes eingeben, dazu zwei verschiedene Stichwörter, die etwas über den Inhalt aussagen (und nicht bereits im Titel vorkommen!), außerdem eine Angabe zur Art des Materials (Text, Schaubild, Karte usw.) und natürlich die Quelle (Zeitschrift, Buch, eigene Sammlung usw.).

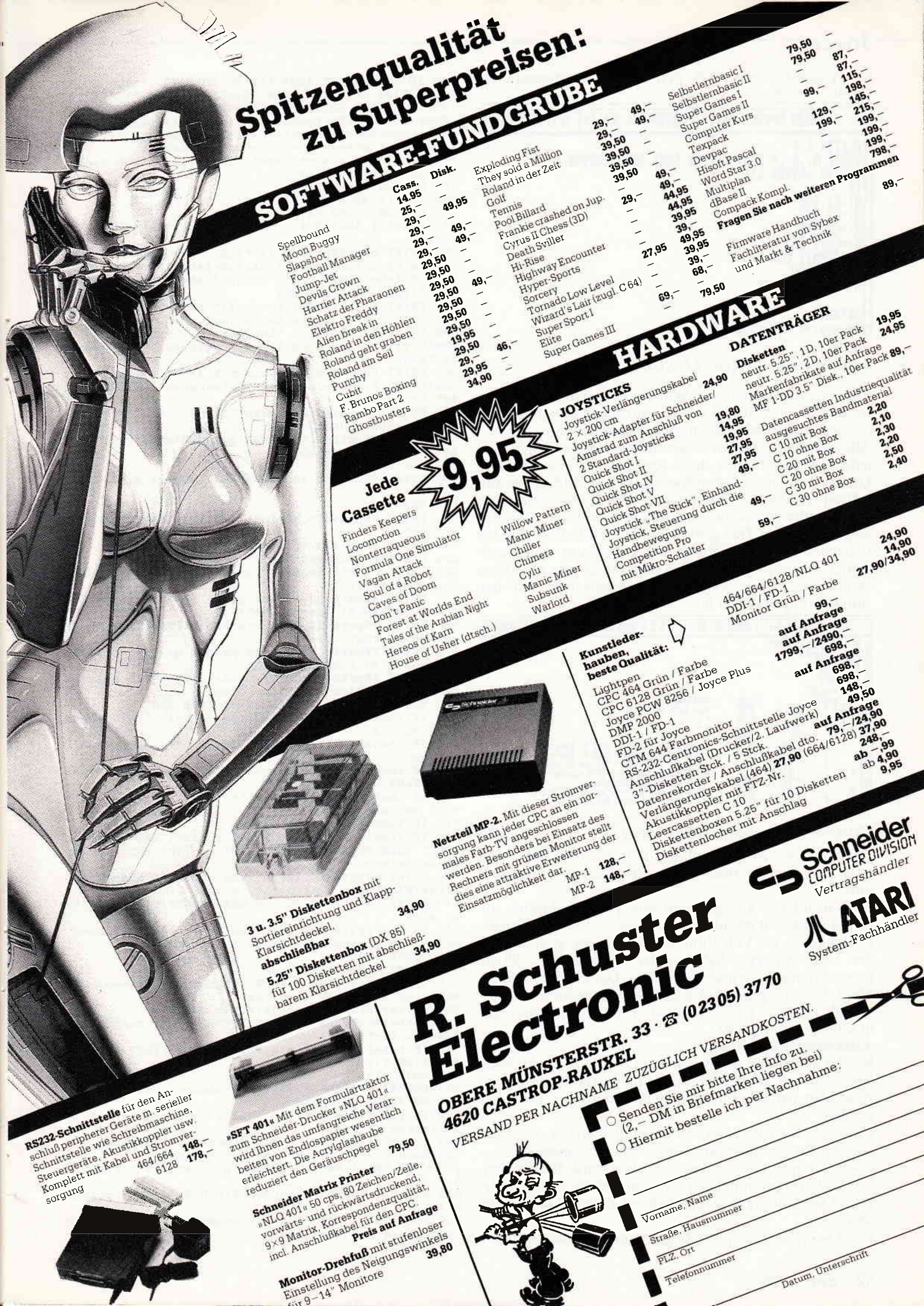
Die Stichwörter werden in einer eigenen Kartei in alphabetischer Reihenfolge gespeichert und können als "Gedächtnisstütze" beim Ausfüllen oder Durchsuchen einer Kartei auf den Bildschirm geholt werden. Dadurch kann man es auch vermeiden, zwei verschiedene, aber trotzdem fast identische Stichwörter zum selben Sachverhalt einzugeben. Sollte ein Titel sehr kurz sein, kann man den Rest der Titelzeile bereits für ergänzende Angaben nutzen!

Schreibfehler oder sonstige Fehler bei der Eingabe können sofort leicht berichtigt werden. Karten, die nicht mehr benötigt werden oder fehlerhaft ausgefüllt wurden, können überschrieben bzw. zunächst einmal gelöscht werden. Dazu muß man den Löschmodus (als erstes Zeichen in der Titelzeile den Stern ("*") eingeben) aufrufen und die Reg.-Nr. der zu löschenden Karteikarte eingeben. Diese wird auf dem Bildschirm ausgegeben, damit sie eventuell als Vorlage für die Neueingabe dienen kann. Will man eine Karte ganz löschen, drückt man in den Eingabezeilen jeweils nur die [Return]-Taste. Dabei empfiehlt es sich allerdings, die zu löschende Karte mit einer Kennung zu versehen (z.B. "&&&" als Titel), um sie später mit dem Suchprogramm wiederfinden zu können.

Mit einem kleinen Trick kann man auch die "Inhaltsangabe" für eine Kartei verändern. Dazu muß man den Löschmodus aufrufen und als Reg.-Nr. die "1" eingeben. Da die Inhaltsangabe einer Kartei genauso lang ist wie die Zeilen Autor/Titel, 1. Stichwort und 2. Stichwort zusammen, schreibt man die neue Inhaltsangabe so in die drei genannten Eingabezeilen, als seien diese zusammenhängend. Unter dem Begriff "Quelle" muß man aber unbedingt wieder die Zahl eintragen, die in der zu überschreibenden Karte steht (dies ist die Nr. der nächsten freien Karteikarte und für das reibungslose Funktionieren verschiedener Programmroutinen verantwortlich!).

Im Unterprogramm "Suchen" kann als Suchkriterium jedes beliebige Wort eingegeben werden. Dabei kann das Suchwort 1 – 14 Zeichen lang sein. In den meisten Fällen dürfte dies ausreichen, um jeden passenden Titel aus der Kartei "herauszufischen". Je kürzer das eingegebene Suchwort ist, umso leichter kann es allerdings auch zu unvermeidbaren "Mißverständnissen" kommen: Das Wort "Eis" steckt halt sowohl in "Eiszeit", als auch in "Eisen"! Beim Suchen wird nämlich jede Karteikarte daraufhin untersucht, ob auf ihr die eingegebene Kombination von Buchstaben und/oder Ziffern vorkommt, wobei Groß-/Kleinschreibung nicht berücksichtigt wird. Das Suchwort "Computer" würde z.B. sowohl den Titel "Com-


RSE Software
(0 23 05) 37 70
Die intelligente Dimension
SOFTWARE-AUTOREN GESUCHT!
Amsmonix
· Komfortabler Monitor-Debugger · 100% Maschinencode · RAM-ROM-AMSDOS-Monitor · Disassembler · Such-Fill-Modifizierfunktion · Laden und Speichern von Files · CAT-Funktion · Intellig. Kopieren (Blockverschiebung) · Registeranzeige · Aufruf von Maschinenprogrammen · Texteingabe · Druckerprotokoll · Anwahl auch von Expansion-Roms · Monitor, ASCII o. Prüfsummen-Anz. · CPC 464, 664, 6128
Cassette 49,– DM* Diskette 59,– DM*
Lagerverwaltung
· 100% Maschinencode · Relative Dateiverwaltung · Professioneller Editor · Bestands-/Inventurliste · Artikel VK-Preisliste · Etikettenausdruck · Ausdruck auf Einzel- o. Endlospapier · Geschützter EK-Preis · Statistik · Integrierte Hardcopyfunktion · Schnelle Sortierroutine · Schnelles Auffinden von Daten · CPC 464, 664, 6128
Diskette 79,– DM*
Mathe-Pack
· Vom Anfänger bis zum Profi · Eingebauter Taschenrechner · Integrierte Hardcopyfunktion · Algebra · Geometrie · Trigonometrie · Mengenlehre · Analysis · Kurvendiskussion · Benutzerfreundlich · CPC 464, 664, 6128
Cassette 59,– DM* Diskette 69,– DM*
Adressverwaltung
· 100% Maschinencode · Relative Dateiverwaltung · Professioneller Editor · Schnelle Sortierroutine · Schnelles Auffinden von Daten · 1 Benutzerfeld frei definierbar · 900 Datensätze · Etikettenausdruck für verschiedene Formate · Listenausdruck mit Einzel- oder Endlosblattunterstützung · Integrierte Hardcopyfunktion · DIN-Tastaturbelegung · CPC 464, 664, 6128
Diskette 59,– DM*
Disc-Scanner
· 100% Maschinencode · Unterstützt 2 Laufwerke · Disketten kopieren · Lesen/Schreiben beliebiger Sektoren · Sektoren modifizieren (Full Screen) · Umrechnen von Blocks in Track/Sektor · Files Umbenennen und Löschen · Reaktivieren gelöschter Files · Graph. Darstellung der Diskbelegung · Usernummern wählbar · Fileinformationen abrufen · Formatieren / opt. Einzelspur Vendor / Data / 43 Tracks · Integrierte Hardcopyfunktion · CPC 464, 664, 6128
Diskette 79,– DM*
Disc-Sorter
· 100% Maschinencode · Relative Dateiverwaltung · Automatische beidseitige Programmerfassung von einer Diskette in allen User-Bereichen · wahlweise manuelle Erfassung · Automatische Erfassung und Verarbeitung aller Fileinformationen · Professioneller Editor · schnelle Sortierroutine · schnelles Auffinden von Daten · Listenausdruck mit Einzel- oder Endlosblattunterstützung · Integrierte Hardcopyfunktion · CPC 464, 664, 6128
Diskette 69,– DM*
Mathe-Pack
Mit diesem Programm halten Sie ein Mathe-Programm in Händen, das sowohl für den Anfänger als auch für den Profi geeignet ist. Es trainiert oder löst Ihre Aufgaben umfassend in: – Algebra – Gleichungssysteme nach Gauß beliebige Gleichungen mit einer Variablen – Geometrie – Trigonometrie – Mengenlehre – Analysis z. B. flexibler Funktionsplotter/4 Funktionen gleichzeitig z. B. Bildung von Ableitungen – Kurvendiskussion z. B. Nullstelle/Extremstelle/Wendestelle/Integration/Wertetabelle
Für einfache Rechenarten ist zusätzlich ein Taschenrechner eingebaut. Auf jeder wichtigen Benutzerebene haben Sie die Möglichkeit einer Hardcopyfunktion. Hierbei können Sie eine Text- und Grafik-Hardcopy ausdrucken lassen. Die Hardcopy ist an verschiedene Druckertypen anpaßbar. Eine benutzerfreundliche Menuesteuerung begleitet Sie durch sämtliche Programmpunkte.
Intelligente Software für intelligente Anwender.
○ Senden Sie mir bitte Ihre Info zu. (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift
Super-Hardcopy
· 100% Maschinencode · RSX-Befehle · Menuegesteuert an versch. Druckertypen anpaßbar · Text-Hardcopyfunktion · Parameter wählbar · Normal- und Inversdruck · komprimierte Hardcopy · 4-Farb-Darstellung möglich · 1:1 Grafik-Hardcopy vom Bildschirm · CPC 464, 664, 6128
Cassette 39,80 DM* Diskette 49,80 DM*
Copy-Master
· Universelles File-Transfer Programm · 100% Maschinencode · Kopiert und analysiert Ihre Software · Cass. zu Cass. · Cass. zu Disc · Disc. zu Cass. · Disc. zu Disc · Disc Backup · Diskettensicherheitskopie auf Cass. (kompl.) · Programmresident im Speicher · Integrierte Hardcopyfunktion · Programmverschiebung bei AMSDOS-Überschreibung · Kopieren v. Headerlosen Programmen · CPC 464, 664, 6128
Cassette 49,– DM* Diskette 59,– DM*
Telecom 1000
· Professionelles DFÜ-Programm · 100% Maschinencode · Einstellen d. Übertragungsparameter · 75, 110, 150, 300, 600, 1200 Baud · Voll- und Halbduplex · 5, 6, 7 oder 8 Datenbits · 1 oder 2 Stopbits · Keine, gerade oder ungerade Parität · versch. Übertragungsprotokolle · Über 32 K Textbuffer · Notizzettel · Professioneller Editor · Übertragung von ASCII, Binär-, Basic-Dateien · Converterprogramm · CPC 464, 664, 6128
Cassette 89,– DM* Diskette 79,– DM*
Assembler-Disassembler
· Professionelles Maschinensprache-Entwicklungspaket · 100% Maschinensprache · Monitor · Assembler (Z 80 Standard) · Disassembler · Debugger (Breakpoint) · Single Stepper (Tracer) · Professioneller Editor · Druckerausgabe · Druckerprotokoll · CPC 464, 664, 6128
Diskette 89,– DM* Diskette 98,– DM*
Universaldatei
· 100% Maschinencode · Relative Dateiverwaltung · je nach Maske und Indexfelder mehrere Tausend Datensätze möglich · bis zu 50 Felder pro Datensatz · Frei definierbare Eingabemaske · Such- und Druckermaske frei erstellbar · Professioneller Editor · Schnelle Sortierroutine · Schnelles Auffinden von Daten · DIN-Tastaturbelegung · Integrierte Hardcopyfunktion · CPC 464, 664, 6128
Diskette 79,– DM*
Vokabeltrainer
· Univers. f. jede Fremdsprache nutzbar · 100 Vokabeln pro Unit · unbegrenzte Unit-Bearbeitung · 2 Lernstufen PP (Prägen/Prüfen) · Einfache Handhabung durch Cursorblocksteuerung · Lernen von Mehrfachzuweisungen möglich (z. B. unregelmäßige Verben) · Prozentuale Ergebnisauswertung, selbst bei Testunterbrechung · Von Pädagogen empfohlen · Deutscher Zeichensatz · CPC 464, 664, 6128
Cassette 59,– DM* Diskette 69,– DM*
Karteikasten
· 100% Maschinencode · Relative Dateiverwaltung · 400 Karten pro Diskettenseite · Beliebige Suchkriterien · Professioneller Editor · Schnelle Sortierroutine · Schnelles Auffinden von Daten · Integrierte Hardcopyfunktionen · Listenausdruck · DIN-Tastaturbelegung · CPC 464, 664, 6128
Diskette 79,– DM*
R. Schuster Electronic
OBERE MÜNSTERSTR. 33
☎ (0 23 05) 37 70
4620 CASTROP-RAUXEL
VERSAND PER NACHNAME ZUZÜGLICH VERSANDKOSTEN. INFO GEGEN 2,– DM IN BRIEFMARKEN.
HÄNDLERANFRAGEN ERWÜNSCHT.
Schneider COMPUTER DIVISION Vertragshändler
* UNVERBINDLICHE PREISEMPFEHLUNG
RSE-SOFTWARE ERHALTEN SIE BEI IHREM FACHHÄNDLER ODER DIREKT BEI
Computer + Bürotechnik R. Kalisch Marktstr. 26, 4353 Oer-Erkenschwick
FERNSEH BERLET Elseyer Str. 12-14, 5800 Hagen-Hohenlimburg
Computerstudio HÖLSCHER Zeppelinstr. 7, 5010 Bergheim
SFK elektro GmbH Delsterner Str. 23, 5800 Hagen
GAI Computer Rathausstr. 28, 7107 Neckarsulm
und bei R. SCHUSTER ELECTRONIC



Spitzenqualität zu Superpreisen:
SOFTWARE-FUNDGRUBE
Cass. Disk.
Spellbound 14.95 –
Moon Buggy 25,– 49,95
Slapshot 29,– –
Football Manager 29,– 49,–
Jump-Jet 29,– 49,–
Devils Crown 29,– –
Harrier Attack 29,50 –
Schatz der Pharaonen 29,50 –
Elektro Freddy 29,50 49,–
Alien break in 29,50 –
Roland in den Höhlen 29,50 –
Roland geht graben 29,50 –
Roland am Seil 19,95 –
Punchy 29,50 46,–
Cubit 29,– –
F. Brunos Boxing 29,95 –
Rambo Part 2 34,90 –
Ghostbusters
Exploding Fist 29,– 49,–
They sold a Million 29,– 49,–
Roland in der Zeit 39,50 –
Golf 39,50 –
Tennis 39,50 –
Pool Billard 39,50 49,–
Frankie crashed on Jup. 29,– 49,–
Cyrus II Chess (3D) – 44,95
Death Sviller – 44,95
Hi-Rise – 39,95
Highway Encounter – 39,–
Hyper-Sports – 49,95
Sorcery 27,95 39,95
Tornado Low Level – 39,–
Wizard's Lair (zugl. C 64) – 68,–
Super Sport I 69,– 79,50
Elite
Super Games III
Selbstlernbasic I 79,50 –
Selbstlernbasic II 79,50 87,–
Super Games I – 87,–
Super Games II – 115,–
Computer Kurs 99,– 198,–
Texpack – 145,–
Devpac 129,– 215,–
Hisoft Pascal 199,– 199,–
Word Star 3.0 – 199,–
Multiplan – 199,–
dBase II – 798,–
Compack Kompl.
Fragen Sie nach weiteren Programmen
Firmware Handbuch 89,–
Fachliteratur von Sybex
und Markt & Technik
Jede Cassette 9,95
Finders Keepers
Locomotion
Nonterraqueous
Formula One Simulator
Vagan Attack
Soul of a Robot
Caves of Doom
Don't Panic
Forest at Worlds End
Tales of the Arabian Night
Hereos of Karn
House of Usher (dtsch.)
Willow Pattern
Manic Miner
Chiller
Chimera
Cylu
Manic Miner
Subsunk
Warlord
HARDWARE
JOYSTICKS
Joystick-Verlängerungskabel 2 × 200 cm 24,90
Joystick-Adapter für Schneider/Amstrad zum Anschluß von 2 Standard-Joysticks 19,80
Quick Shot II 14,95
Quick Shot IV 19,95
Quick Shot V 27,95
Quick Shot VII 27,95
Joystick „The Stick", Einhand-Joystick, Steuerung durch die Handbewegung 49,–
Competition Pro mit Mikro-Schalter 49,–
59,–
DATENTRÄGER
Disketten
neutr. 5.25", 1D, 10er Pack 19,95
neutr. 5.25", 2D, 10er Pack 24,95
Markenfabrikate auf Anfrage
MF 1-DD 3.5" Disk., 10er Pack 89,–
Datencassetten Industriequalität ausgesuchtes Bandmaterial
C 10 mit Box 2,20
C 10 ohne Box 2,10
C 20 mit Box 2,30
C 20 ohne Box 2,20
C 30 mit Box 2,50
C 30 ohne Box 2,40
Kunstlederhauben, beste Qualität:
464/664/6128/NLQ 401 24,90
DDI-1 / FD-1 14,90
Monitor Grün / Farbe 27,90/34,90
Lightpen 99,–
CPC 464 Grün / Farbe auf Anfrage
CPC 6128 Grün / Farbe auf Anfrage
Joyce PCW 8256 / Joyce Plus 1799,–/2490,–
DMP 2000 698,–
DDI-1 / FD-1 auf Anfrage
FD-2 für Joyce 698,–
CTM 644 Farbmonitor 698,–
RS-232-Centronics-Schnittstelle Joyce 148,–
Anschlußkabel (Drucker/2. Laufwerk) 49,50
3"-Disketten Stck. / 5 Stck. auf Anfrage
Datenrekorder / Anschlußkabel dto. 79,–/24,90
Verlängerungskabel (464) 27,90 (664/6128) 37,90
Akustikkoppler mit FTZ-Nr. 248,–
Leercassetten C 10 ab –,99
Diskettenboxen 5.25" für 10 Disketten ab 4,90
Diskettenlocher mit Anschlag 9,95
Netzteil MP-2. Mit dieser Stromversorgung kann jeder CPC an ein normales Farb-TV angeschlossen werden. Besonders bei Einsatz des Rechners mit grünem Monitor stellt dies eine attraktive Erweiterung der Einsatzmöglichkeit dar.
MP-1 128,–
MP-2 148,–
3 u. 3.5" Diskettenbox mit Sortiereinrichtung und Klappsichtdeckel, abschließbar 34,90
5.25" Diskettenbox (DX 85) für 100 Disketten mit abschließbarem Klarsichtdeckel 34,90
Schneider COMPUTER DIVISION Vertragshändler
ATARI System-Fachhändler
R. Schuster Electronic
OBERE MÜNSTERSTR. 33 · ☎ (023 05) 37 70
4620 CASTROP-RAUXEL
VERSAND PER NACHNAME ZUZÜGLICH VERSANDKOSTEN.
RS232-Schnittstelle für den Anschluß peripherer Geräte m. serieller Schnittstelle wie Schreibmaschine, Steuergeräte, Akustikkoppler usw. Komplett mit Kabel und Stromversorgung
464/664 148,–
6128 178,–
»SFT 401« Mit dem Formulartraktor zum Schneider-Drucker »NLQ 401« wird Ihnen das umfangreiche Verarbeiten von Endlospapier wesentlich erleichtert. Die Acrylglashaube reduziert den Geräuschpegel 79,50
Schneider Matrix Printer »NLQ 401« 50 cps, 80 Zeichen/Zeile, vorwärts- und rückwärtsdruckend, 9×9 Matrix, Korrespondenzqualität, incl. Anschlußkabel für den CPC. Preis auf Anfrage
Monitor-Drehfuß mit stufenloser Einstellung des Neigungswinkels für 9–14" Monitore 39,80
○ Senden Sie mir bitte Ihre Info zu. (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme:
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift

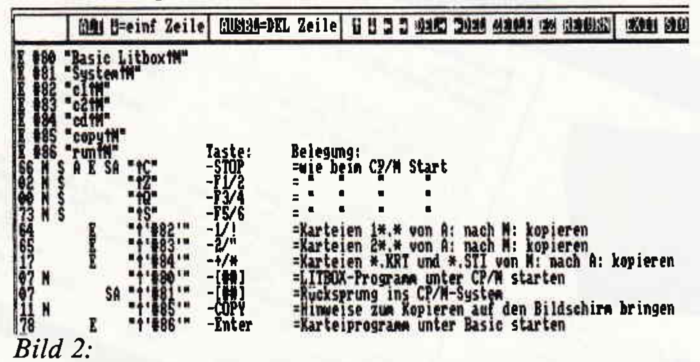
putergraphik", als auch den Titel "Microcomputer" heraussuchen.



```
PALETTE 1 0
SETDEF M:,A:,* ÄORDER=(SUB,COM) TEMPORARY=M:Ü
SETKEYS TASTEN.LIT
PIP
<M:=C1.SUB
<M:=C2.SUB
<M:=CD.SUB
<M:=COPY.SUB
<M:=SUBMIT.COM
<M:=PIP.COM
<M:=HINWEISE.LIT
<
↑AE↑ÄH↑M
pip con:=M:HINWEISE.LIT
```

Bild 1:
Auto-Start-Datei PROFILE.SUB

Zum Suchen können zwei Wörter gleichzeitig eingegeben werden. In diesem Fall werden nur Karteikarten aufgelistet, auf denen beide Suchwörter vorkommen.
Alle vom Programm zu einem Suchwort gefundenen Karteikarten werden mit ihrer Reg.-Nr. versehen, und in Gruppen von max. sieben Karten auf dem Bildschirm aufgelistet. Zu Dokumentationszwecken können sie dann einzeln oder gruppenweise ausgedruckt werden. Hat man einen gesuchten Titel gefunden, kann man die Suche vor Erreichen des Kartei-Endes abbrechen, um neue Suchwörter einzugeben. Als Hilfe kann auch hier eine Liste sämtlicher bisher eingegebenen Stichwörter auf den Bildschirm geholt werden.



```
E #80 "Basic Litbox↑M"
E #81 "System↑M"
E #82 "c1↑M"
E #83 "c2↑M"
E #84 "cd↑M"
E #85 "copy↑M"
E #86 "run↑M"                 Taste:    Belegung:
66 N S A E SA "↑C"            -STOP     =wie beim CP/M Start
02 N S        "↑Z"            -F1/2     =  "    "    "     "
00 N S        "↑Q"            -F3/4     =  "    "    "     "
73 N S        "↑S"            -F5/6     =  "    "    "     "
64       E    "↑'#82'"        -1/!      =Karteien 1*.* von A: nach M: kopieren
65       E    "↑'#83'"        -2/"      =Karteien 2*.* von A: nach M: kopieren
17       E    "↑'#84'"        -+/*      =Karteien *.KRT und *.STI von M: nach A: kopieren
07 N          "↑'#80'"        -[##]     =LITBOX-Programm unter CP/M starten
07         SA "↑'#81'"        -[##]     =Rücksprung ins CP/M-System
11 N          "↑'#85'"        -COPY     =Hinweise zum Kopieren auf den Bildschirm bringen
78       E    "↑'#86'"        -Enter    =Karteiprogramm unter Basic starten
```

Bild 2:
Tastendefinitionsdatei TASTEN.LIT

Während des Programmlaufs kann selbstverständlich zwischen Suchen und Ergänzen jederzeit gewechselt werden.
Auf Grund des angegebenen Suchprinzips könnte die Kartei auch sehr gut als Adressenverzeichnis, als Schallplattenkartei, als Videofilmverzeichnis usw. genutzt werden. Dazu müßten lediglich in den Zeilen 2550 - 2590 die Bezeichnungen verändert werden. Dabei sollten allerdings die vorgegebenen Längen der Bezeichnungen beibehalten werden, um die "Umbauarbeiten" am Programm in Grenzen zu halten.

Anmerkungen zu Listing 1:

Bei der Erstellung des Programms wurde versucht, ohne Rücksicht auf den Speicherplatz die Variablennamen so zu wählen, daß ihre Funktionen im Programm erkennbar werden.
Um das Lesen des Listings zu erleichtern, wurden die meisten Programmsprünge mit einer "Zielangabe" (----- >>> Zielangabe) versehen. Außerdem wurde bei den meisten Unterprogrammen, zu denen mit einem GOSUB verzweigt wird, angegeben, aus welcher Zeile sie aufgerufen werden.
Das Listing wurde mit 50 Zeichen pro Zeile ausgedruckt.

```
440 IF taste$="2" THEN LPRINT CHR$(27);"c";
450 LPRINT CHR$(27);CHR$(15);CHR$(27);"1";CHR$(10)
;CHR$(27);"Q";CHR$(100);CHR$(27);"C";CHR$(60);'-->
 Kompakt-Schrift/Randeinstellung/Zeilenzahl
460 '==================== Suchen geeigneter Karte
ien/Menü 1 ====================
470 PRINT cl$;c.aus$;FNpkt$(15,32,inv.an$+" Einen
Moment... "+inv.aus$)
480 OPTION FILES "M"
490 inhalt.M$=FIND$("*.KRT")
500 OPTION FILES "A"
510 inhalt1.A$=FIND$("1*.KRT"):inhalt2.A$=FIND$("2
*.KRT")
520 PRINT cl$;FNpkt$(1,14,"Auf diesen Disketten be
finden sich folgende LITERATUR-KARTEIEN:")
530 IF inhalt.M$="" AND inhalt1.A$="" AND inhalt2.
A$="" THEN 540 ELSE 550
540 PRINT FNpkt$(4,41,inv.an$+" keine "+inv.aus$):
GOTO 570'-----> keine Kartei gefunden
550 PRINT FNpkt$(3,25,"Laufwerk M: "+inv.an$+" "+i
nhalt.M$+" "+inv.aus$)'-----> gefundene Karteien a
uflisten
560 PRINT FNpkt$(5,25,"Laufwerk A: "+inv.an$+" "+i
nhalt1.A$+" "+inv.aus$+"    "+inv.an$+" "+inhalt2.A
$+" "+inv.aus$)
570 PRINT inv.aus$:sw.neu$=""'-----> Prüfvariable
für Eingaben zurücksetzen
580 PRINT FNpkt$(8,15,CHR$(150)+STRING$(60,154)+CH
R$(156))'-----> Rahmen
590 FOR x=9 TO 15
600 PRINT FNpkt$(x,15,CHR$(149));FNpkt$(x,76,CHR$(
149))
610 NEXT
620 PRINT FNpkt$(16,15,CHR$(147)+STRING$(60,154)+C
HR$(153))
630 PRINT FNpkt$(9,18,inv.an$+" d "+inv.aus$+" And
ere Diskette durchsuchen")
640 PRINT FNpkt$(10,22,"(vor der Programmwahl die
Disk. wechseln!)")
650 PRINT FNpkt$(12,18,inv.an$+" b "+inv.aus$+" Vo
rhandene KARTEI bearbeiten")
660 PRINT FNpkt$(14,18,inv.an$+" e "+inv.aus$+" Ne
ue KARTEI erstellen")
670 PRINT FNpkt$(15,22,"(Es passen maximal 2 KARTE
IEN auf eine Disk.-Seite!)")
680 PRINT FNpkt$(18,27,inv.an$+" Bitte die entspre
chende Taste drücken "+inv.aus$)
690 taste$=INKEY$:IF taste$="" THEN 690
700 IF taste$=CHR$(8) THEN 130'----->>> Zur Startm
aske zurück
710 IF taste$=CHR$(27) THEN 3640'-----> Programm b
eenden
720 IF taste$="d" OR taste$="b" OR taste$="e" THEN
 730 ELSE 690
730 IF taste$="e" THEN 1980'----->>> eine leere Ka
rtei neu einrichten
740 IF taste$="d" THEN 470
750 IF inhalt.M$="" AND inhalt1.A$="" AND inhalt2.
A$="" THEN PRINT cl$:GOTO 470'----->>> wdh., da Ei
ngabe falsch
760 PRINT FNpkt$(20,16,"Welche der oben angegebene
n KARTEIEN soll bearbeitet werden?");
770 PRINT FNpkt$(22,38,inv.an$+SPACE$(9)+".KRT "+i
nv.aus$)
780 PRINT FNpkt$(22,39,""+inv.an$);:laenge.wort=8'
-----> Cursorpos./Wortlänge festlegen
790 GOSUB 3290'----->>> UP Texteingabe
800 PRINT c.aus$;inv.aus$
810 IF zchn$<>CHR$(8) THEN 820 ELSE 470'----->>> w
eiter od. zum Menü 1 zurück
820 wort$=LEFT$(wort$+STRING$(8,32),8)
830 kartei$=UPPER$(wort$+".KRT"):stichw.kartei$=UP
PER$(wort$+".STI")
840 IF inhalt.M$=kartei$ THEN OPTION FILES "M" ELS
E OPTION FILES "A"
850 GOTO 870
860 RESUME 770'-----> Eingabe wiederholen
870 ON ERROR GOTO 860
880 IF FIND$(kartei$)="" THEN 470'--->>> Menü 1, d
a Eingabe falsch
890 '==================== angegebene Karteien öff
nen/Menü 2 ============================
900 OPEN "R",1,Kartei$:OPEN "R",2,stichw.kartei$,1
4
910 FIELD 1,128 AS d.karte$:FIELD 2,14 AS d.stichw
ort$
920 FOR x=1 TO 108'-----> Stichwörterkartei in Var
iablentabelle übertragen
930 GET 2
```


ÜBER 380 SCREENS
Mit blitzschnellem Bildschirmaufbau.
In futuristischem High-Tech-Design.
SPINDIZZY ist eine künstliche Welt, ein riesiges Puzzle aus steilen Abhängen, engen Korridoren, hohen Wänden, schmalen Kanten und plötzlich auftauchenden Schluchten, durch die man seinen Weg finden muß. Mit System und ohne herunterzupurzeln.
Sinn und Zweck der Übung:
Die Abteilung für Geheimprojekte hat eine künstliche Welt entdeckt, die irgendwo im Raum schwebt. Und Ihre Aufgabe ist es nun, eine exakte Landkarte anzufertigen. Allerdings verfügen Sie nur über ein sehr altes Raumschiff, das ungeheuer viel Energie verbraucht, so daß Sie darauf angewiesen sind, sich auf dem künstlichen Planeten selbst zu versorgen.
Ein harter Job unter Zeitdruck, bei dem man häufig auf schier unüberwindliche Hindernisse trifft.
Und: Die Lösung liegt im 386. Screen!
Ein besonderer Leckerbissen: Die
MULTI-PERSPECTIVE
Funktion, mit der man die Hindernisse aus verschiedenen Blickwinkeln betrachten kann.
Spindizzy
A 3-D GAME OF SKILL
BY PAUL SHIRLEY
AMSTRAD®/SCHNEIDER®CASSETTE
Electric Dreams
SOFTWARE
Electric Dreams
Erhältlich als Cassette + Diskette für Schneider CPC und Commodore 64, als Cassette für Sinclair Spectrum.
Activision Deutschland GmbH, Postfach 76 06 80, 2000 Hamburg 76.
VERTRIEB DEUTSCHLAND:
Ariolasoft (Exclusiv-Distributor)
Rushware (Autorisierter Mitvertrieb)
VERTRIEB ÖSTERREICH:
Karasoft (Exclusiv-Distributor)
VERTRIEB SCHWEIZ
HILCU (Exclusiv-Distributor)

```
      |[ALT] U=einf Zeile|[AUSSCHN]=DEL Zeile| ...
erase M:*.KRT
<y
erase M:*.STI
<y
pip
<M:=A:1*.*
<con:=M:Hinweise.Lit
<
```

Kopierprogramm C1.SUB Kartei 1.* von A: nach M: kopieren, eine in M: schon existierende Literaturkartei wird vorher gelöscht.* *Bild 3*

```
      |[ALT] U=einf Zeile|[AUSSCHN]=DEL Zeile| ...
erase M:*.KRT
<y
erase M:*.STI
<y
pip
<M:=A:2*.*
<con:=M:Hinweise.Lit
<
```

Kopierprogramm C2.SUB Kartei 2.* von A: nach M: kopieren, wie Bild 3* *Bild 4*

Bei der Eingabe des Programms ist es sinnvoll, die Weite des Bildschirms mit WIDTH 50 ebenfalls auf diese Breite einzustellen. Danach wird mit WIDTH 90 die "normale" Breite wieder eingestellt.

Anmerkungen zu einigen Details des Programms:

Z. 110:
Die Funktion der STOP-Taste wird hier ausgeschaltet, weil ein STOP an der falschen Stelle während des Programmlaufs zu einem Teilverlust neu eingegebener Daten führen könnte.

Z. 450:
Möchte man eine andere Schrift- und Randeinstellung für den Drucker haben, müssen die ESC-(CHR$-)-Folgen in dieser Zeile geändert werden (vgl. dazu das Joyce-Handbuch 1, Kap. II.3 und II.6).

Z. 490:
Als Typenbezeichnung für die Dateien wurde *.KRT bzw. *.STI gewählt. Das Programm beachtet bei allen Diskettenoperationen nur Dateien mit diesen Typenbezeichnungen. Daher sollte darauf geachtet werden, daß auf den Disketten

```
940 sw$(x)=d.stichwort$
950 NEXT
960 sw$(0)=STRING$(14,32)'-----> niederwertigstes
Wort für Sortierprogramm
970 sw$(109)=STRING$(14,127)'--> höchstwertigstes
Wort
980 GET 1,1'-----> Informationen zur Kartei lesen
990 nr.letzte=VAL(MID$(d.karte$,94,5))-1'-----> Nr
. der letzten beschriebenen Karte
1000 '------------------------------ Bildschirmmask
e ------------------------------------
1010 PRINT cl$;c.aus$;inv.aus$
1020 PRINT FNpkt$(8,28,"Inhalt der KARTEI "+inv.an
$+" "+kartei$+": "+inv.aus$)
1030 PRINT FNpkt$(10,3,inv.an$+" "+LEFT$(d.karte$,
83)+inv.aus$)'---> Info. über die Kartei
1040 PRINT FNpkt$(13,29,CHR$(150)+STRING$(26,154)+
CHR$(156))
1050 FOR x=14 TO 18
1060 PRINT FNpkt$(x,29,CHR$(149));FNpkt$(x,56,CHR$
(149))
1070 NEXT
1080 PRINT FNpkt$(19,29,CHR$(147)+STRING$(26,154)+
CHR$(153))
1090 PRINT FNpkt$(15,32,inv.an$+" s "+inv.aus$+" K
ARTEI durchsuchen")
1100 PRINT FNpkt$(17,32,inv.an$+" e "+inv.aus$+" K
ARTEI ergänzen")
1110 PRINT FNpkt$(22,24,inv.an$+" Bitte die entspr
echende Taste drücken "+inv.aus$)
1120 PRINT c.aus$
1130 taste$=INKEY$:IF taste$="" THEN 1130
1140 IF taste$=CHR$(8) THEN GOSUB 3650:GOSUB 3630:
GOTO 470'----->>> Programmablauf zurücksetzen
1150 IF taste$=CHR$(27) THEN 3640'-----> Programm
beenden
1160 IF taste$="s" THEN 1200 '---->>> UP Suchen
1170 IF taste$="e" THEN 2430 ELSE 1120 '----->>> z
ur Eingaberoutine bzw. Wiederholung
1180 '=============== HP Durchsuchen der Kartei (
Einspr. aus 1160) =====================
1190 '------------------------------ Bildschirmmask
e ------------------------------------
1200 PRINT cl$
1210 PRINT FNpkt$(0,0,inv.an$+" SUCHEN: "+inv.aus$
+" "+kartei$)
1220 PRINT FNpkt$(0,24,inv.an$+" Einträge:"+STR$(n
r.letzte)+" "+inv.aus$)
1230 PRINT FNpkt$(5,0,STRING$(89,138))
1240 PRINT FNpkt$(7,10,inv.an$+" ? "+inv.aus$+" Au
flisten der bisher gespeicherten Stichwörter")
1250 PRINT FNpkt$(9,6,">>> Werden zwei Suchbegriff
e genannt, so werden nur Karteikarten")
1260 PRINT FNpkt$(10,10,"ausgegeben, in denen beid
e Begriffe gemeinsam vorkommen")
1270 PRINT FNpkt$(12,6,">>> Wird kein Suchbegriff
```

Abo-Order
»ZEITSCHRIFT«
»DATABOX«

Abgesandt am:

______________ 198___

Bemerkungen:

____

____

____

____



Abo-Order
»ZEITSCHRIFT«
»DATABOX«

Das kompetente Magazin für alle Schneider-Anwender!

Bestellen Sie noch heute Ihr Abonnement mit dieser Postkarte!

Bitte ausreichend frankieren

Antwortkarte

Schneider CPC International
Postfach 250

3440 Eschwege

»RESTPOSTEN-PACK 1985«
»Einzelhefte 1986«

Abgesandt am:

______________ 198___

Bemerkungen:

____

____

____

____



SONDERAKTION
»Restposten-Pack 1985«
»Einzelheftbestellung 1986«

Absender: *(Bitte genaue Anschrift angeben!)*

____
Name

____
Vorname

____
Firma

____
Straße/Nr./Postfach

____
PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

Schneider CPC International
Postfach 250

3440 Eschwege

Einzelbezug
»DATABOX«

Abgesandt am:

______________ 198___

Bemerkungen:

____

____

____

____



Einzelbezug
»DATABOX«

Absender: *(Bitte genaue Anschrift angeben!)*

____
Name

____
Vorname

____
Firma

____
Straße/Nr./Postfach

____
PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

Schneider CPC International
Postfach 250

3440 Eschwege

»Abo-Order Zeitschrift«

Hiermit bestelle ich „Schneider CPC International" für mindestens
☐ 1/2 Jahr ☐ 1 Jahr

Zustellung, Vertriebskosten und MwSt. sind im günstigen Preis von 30,- DM bzw. 60,- DM enthalten (BRD und West-Berlin).
Auslandspreise: **Europa** 12 Ausgaben – 90,- DM,
6 Ausgaben – 45,- DM
Außereuropäisches Ausland 12 Ausgaben – 120,- DM,
6 Ausgaben – 60,- DM
Lieferung soll ab Ausgabe Nr. .... erfolgen.

Lieferanschrift
☐ Geschäft ☐ Privat

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 1/2 bzw. 1 Jahr, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum Unterschrift
(bei Minderjährigen: Unterschrift des ges. Vertreters)

Bitte unbedingt zwei Unterschriften leisten!

»Abo-Order DATABOX«

Hiermit bestelle ich die ☐ CPC Databox ☐ Joyce-Databox
für mindestens ☐ 1/2 Jahr ☐ 1 Jahr
als ☐ Cassette ☐ Diskette.

Bezugspreise Inland (einschl. Porto/Verpackung)
Cass. 1/2 Jahr 90,- DM, 1 Jahr 180,- DM
Disk. 3" 1/2 Jahr 150,- DM, 1 Jahr 300,- DM
Bezugspreise Ausland (eingeklammerte Werte außereuropäisch)
Cass. 1/2 Jahr 100,- (120,-) DM, 1 Jahr 200,- (240,-) DM
Disk. 3" 1/2 Jahr 160,- (180,-) DM, 1 Jahr 320,- (360,-) DM

Gewünschte Zahlungsweise:
☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl (von Scheck abschreiben)

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** – zahlbar innerhalb zwei Wochen nach Erhalt
(Bitte keine Vorauszahlung leisten – Rechnung abwarten.)

Garantie:
Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche bei der Bestelladresse schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum Unterschrift

Abo-Zeitschrift
Abo-DATABOX

Bestellen Sie Ihr Abonnement mit der nebenstehenden Karte!
Preisvorteil gegenüber dem Einzelbezug!
Mit einem Databox-Abo erhalten Sie die Software zum Magazin immer pünktlich und mit einem saftigen Preisvorteil gegenüber dem Einzelbezug!

Sonderaktion Restposten-Pack 1985

☐ Aus Restbeständen des Jahrgangs 1985 der Zeitschrift „Schneider CPC International" bestelle ich hiermit die noch vorhandenen vier Ausgaben
8/85 10/85 11/85 12/85
im Paket zum günstigen Gesamtpreis von nur 12,– DM (Ausland 15,– DM). In diesem Preis sind die Kosten für Porto und Verpackung bereits enthalten. Lieferung ist nur per Vorkasse möglich (V-Scheck bitte beifügen).
Nur solange der Vorrat reicht!

Jahrgang 1986

Die Sonderhefte sowie sämtliche Ausgaben von „Schneider CPC International" des Jahrgangs 1986 sind über den Verlag zu beziehen. Bei einem Bestellwert von mindestens 15,– DM werden keine Porto- und Versandgebühren erhoben; bei einem Bestellwert unter 15,– DM werden 3,– DM Porto/Verpackung berechnet (Ausland 5,– DM Porto/Verpackung). Lieferung nur gegen Vorkasse (V-Scheck).
Bestellung 1986'er Ausgaben (bitte ankreuzen):

☐ 1/86	5,– DM
☐ 2/86	5,– DM
☐ 3/86	5,– DM
☐ 4/86	5,– DM
☐ 5/86	6,– DM
☐ Sonderheft 1/86	14,– DM
☐ Sonderheft 2/86	14,– DM
Porto/Verpackung (nur bei einem Bestellwert unter 15,– DM)	DM
Gesamtbetrag	DM

☐ Einen Verrechnungsscheck in Höhe des Rechnungsbetrages habe ich beigefügt.

Datum Unterschrift (bei Minderjährigen des ges. Vertreters)

Bestellkarte
»RESTPOSTEN-PACK 1985«
»EINZELHEFTE 1986«

Wir räumen unser Lager und bieten unseren Lesern, die 1985 noch nicht "dabei" waren, die Möglichkeit, die noch vorhandenen Ausgaben des vergangenen Jahres preisgünstig "im Paket" zur erstehen. Dies sind die Ausgaben 8, 10, 11 und 12/85.
Bitte benutzen Sie hierfür die nebenstehende Karte!
Die 1986'er Ausgaben sind noch sämtlich erhältlich.

»Databox & Databox Highlights
Einzelbestellung

Ich bestelle hiermit die Schneider Databox (bitte ankreuzen):

CPC	Kassette	Diskette 3"
Ausgabe 8/85	☐ 14,– DM	
Ausgabe 9/85	☐ 14,– DM	
Ausgabe 10/85	☐ 14,– DM	
Ausgabe 11/85	☐ 14,– DM	
Ausgabe 12/85	☐ 14,– DM	
Ausgabe 1/86	☐ 14,– DM	
Ausgabe 2/86	☐ 14,– DM	2/86 ☐ 24,– DM
Ausgabe 3/86	☐ 14,– DM	3/86 ☐ 24,– DM
Ausgabe 4/86	☐ 14,– DM	4/86 ☐ 24,– DM
Ausgabe 5/86	☐ 14,– DM	5/86 ☐ 24,– DM
Ausgabe 6/86	☐ 14,– DM	6/86 ☐ 24,– DM
Highlights 1	☐ 14,– DM	
Highlights 2	☐ 14,– DM	
NEU		
Joyce 5/86		5/86 ☐ 24,– DM
Joyce 6/86		6/86 ☐ 24,– DM

+ Porto/Verpackung
(Porto/Verp. Inland 3,– DM)
(Porto/Verp. Ausland 5,– DM) DM

Gesamtbetrag DM

☐ Diesen Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.
☐ Ich bitte um Lieferung per Nachnahme (nur im Inland).
Bei Nachnahme kommt zum o.g. Betrag noch die Nachnahmegebühr hinzu.

Datum Unterschrift (bei Minderjährigen des gesetzl. Vertreters)

DATABOX – jetzt auch im Abonnement erhältlich.
Bitte Bestellkarte benutzen!

»DATABOX«
Einzelbezug

Die Software zum Magazin!
Problemlos bestellen mit der Anforderungskarte!

```
eingegeben (stattdessen 2x <RETURN> drücken),")
1280 PRINT FNpkt$(13,10,"dann werden nacheinander
alle Karteikarten aufgelistet")
1290 PRINT FNpkt$(15,6,">>> Groß-/Kleinschreibung
wird beim Suchen")
1300 PRINT FNpkt$(16,10,"nicht berücksichtigt")
1310 GOTO 1330
1320 PRINT FNpkt$(5,90,"");cl.u$;
1330 PRINT FNpkt$(2,24,"1. Suchbegriff: "+inv.an$+
SPACE$(16)+inv.aus$);
1340 PRINT FNpkt$(4,0,SPACE$(15)+"und/oder 2. Such
begriff: "+inv.an$+SPACE$(16)+inv.aus$)
1350 '----------------------- Eingabe der Suchbeg
riffe ----------------------------------
1360 FOR x=1 TO 7:karte$(x)="":NEXT
1370 PRINT FNpkt$(2,41,""+inv.an$+c.an$);:laenge.w
ort=14'-----> Cursorpos./Wortlänge festlegen
1380 GOSUB 3290'----->>> UP Texteingabe
1390 IF zchn$<>CHR$(8) THEN suchwort1$=UPPER$(wort
$) ELSE 980'-----> weiter oder Abbruch/Rücksprung
1400 IF suchwort1$="" THEN suchwort1$=" "'-----> l
eeren String in 1 Leerzeichen umwandeln
1440 PRINT FNpkt$(4,41,"");:laenge.wort=14'-----/
Cursorpos./Wortlänge festlegen
1450 GOSUB 3290'----->>> UP Texteingabe
1460 IF zchn$<>CHR$(8) THEN suchwort2$=UPPER$(wort
$) ELSE 980'----->>> weiter od. Abbruch/Rückspr.
1470 IF suchwort2$="" THEN suchwort2$=" "'----> le
eren String in 1 Leerzeichen umwandeln
1480 PRINT FNpkt$(4,0,inv.an$+"lfd.Nr."+inv.aus$)
1490 nr=1:PRINT inv.aus$
1500 PRINT FNpkt$(5,90,"");:PRINT cl.u$'---> Curso
rpos./lösch.
1510 FOR nr.karte=2 TO nr.letzte'-----> Karteikart
en sequentiell lesen
1520 GET 1,nr.karte
1530 karte$=d.karte$
1540 GOSUB 1570
1550 NEXT
1560 GOSUB 1700:GOTO 1320'-----> erst zum UP Druck
en, dann zur Suchbegriffeingabe
1570 '------ UP Karteikarten durchsuchen und Bilds
chirmausgabe (Einspr. aus 1540) --------
1580 IF INSTR(UPPER$(karte$),suchwort1$)<>0 THEN 1
590 ELSE RETURN'------> falls 1. Suchbegriff gefun
den, weiter, sonst Rücksprung
1590 IF INSTR(UPPER$(karte$),suchwort2$)=0 THEN RE
TURN'->>> falls 2.Suchwort nicht gefund.,Rücksp.
1600 karte$(nr)=karte$+STRING$(25,32)+"Reg.Nr.: "+
STR$(nr.karte)'---> Karte mit lfd. Nr. versehen u.
"Reg.Nr." anhängen
1610 '---------------- Bildschirmausgabe der gefund
enen Karteikarten ---------------------
1620 PRINT nr;TAB(5)LEFT$(karte$(nr),55);CHR$(133)
;MID$(karte$(nr),56,14);" ";MID$(karte$(nr),70,14)
1630 PRINT TAB(5)MID$(karte$(nr),84,10),MID$(karte
$(nr),94,75);
1640 PRINT STRING$(89,"-");
1650 nr=nr+1'-----> lfd. Nr. erhöhen
1660 IF nr>7 THEN GOSUB 1700 ELSE RETURN'-----> fa
lls 7 Titel ausgegeben, zum UP Drucken
1670 FOR x=1 TO 7:karte$(x)="":NEXT'-----> löschen
1680 RETURN
1690 '================ UP Drucken (Einspr. aus 15
60 oder 1660) ========================
1700 IF karte$(1)="" THEN RETURN'-----> Rückspr.,
da keine zu druckende Karte vorhanden
1710 PRINT FNpkt$(26,0,STRING$(89,138));
1720 '----------------------- Menü für Druckersteu
erung ----------------------------------
1730 PRINT FNpkt$(28,0,inv.an$+" a "+inv.aus$+" al
le Karten drucken   "+inv.an$+" 1,2,...,7 "+inv.au
s$+" (=lfd.Nr.) einzelne Karten drucken"+SPACE$(18
))
1740 PRINT FNpkt$(30,0,inv.an$+" RETURN "+inv.aus$
+" weiter...      "+inv.an$+" CAN "+inv.aus$+" S
uche abbrechen/neues Suchwort"+c.aus$);
1750 taste$=INKEY$:IF taste$="" THEN 1750
1760 IF taste$>"0" AND taste$<"8" OR taste$=CHR$(8
) OR taste$=CHR$(13) OR taste$=CHR$(27) OR taste$=
"a" THEN 1770 ELSE 1750
1770 IF taste$=CHR$(8) THEN 1320'------>>> Suche a
bbrechen/neues Suchwort
1780 IF taste$=CHR$(13) THEN 1960'----->>> Suche f
ortsetzen
1790 IF taste$=CHR$(27) THEN 3640'----->>> Program
m beenden
1800 IF taste$="a" THEN 1890'----->>> alle Karten
drucken
```

mit Literaturkarteien keine umfangreichen anderen Dateien abgespeichert werden! (2x78 kByte = 156 kByte).

```
U=einf Zeile   =DEL Zeile
pip
<A:=M:*.KRT
<A:=M:*.STI
<con:=M:Hinweise.Lit
```

Kopierprogramm CD.SUB Akt. Kartei in M: nach A: kopieren, entspr. Kartei (.* oder 2*.*) in A: wird überschrieben, falls vorhanden.* *Bild 5*

Z. 870/2170:
Die 'ON ERROR GOTO'-Zeilen sollten zunächst nur als REM-Zeilen in das Programm geschrieben werden, da es sonst bei Probeläufen zu keinen verwertbaren Fehlermeldungen kommt.

Z. 900:
Die Karteien sind als Random-Dateien angelegt, damit sie sequentiell durchsucht und direkt geändert werden können.

Z. 920 - 970:
Nachdem eine Kartei zur Bearbeitung geöffnet worden ist, wird zunächst die dazugehörige Stichwörterkartei in eine Variablentabelle (Feld) übertragen. Dadurch kann sie zum einen während des Programmlaufs ziemlich schnell auf den Bildschirm geholt werden, zum anderen werden bei der Eingabe neuer Stichwörter diese sofort alphabetisch in die Tabelle einsortiert (Z. 3210 - 3280). Erst am Ende der Bearbeitung wird diese Tabelle wieder in die Kartei übertragen (Z. 3670 - 3720).

Z. 930 - 940:
Die 1. Karte einer Hauptkartei (NAME.KRT) enthält die Informationen zu dieser Kartei (Inhalt der Kartei und Nr. der nächsten zu beschreibenden Karte.

Z. 1200 - 1300:
Die Informationen zur Programmsteuerung werden nur beim ersten Aufruf des Suchmenues auf dem Bildschirm gezeigt (das spart Zeit).

Z. 1510:
Die Kartei wird immer nur bis zur letzten beschriebenen Karteikarte durchsucht. Diese Nr. steht auf der ersten Karte jeder Hauptkartei und wird bei jeder Eingabe aktualisiert (Z. 3020 - 3030).

```
U=einf Zeile   =DEL Zeile
pip con:=M:Hinweise.Lit
```

Programm COPY.SUB Ausgabe der Datei HINWEISE .LIT, Aufruf durch Betätigen der COPY-Taste. *Bild 6*

Z. 1600:
Die Reg.-Nr. ist die Nummer der aktuellen Karteikarte. Sie wird für die Bildschirm- und Druckerausgabe an die Karteikarte angehängt, weil sie zum Löschen oder Überschreiben einer Karte benötigt wird.

Z. 2180:
Wird beim Erstellen einer neuen (leeren) Kartei ein fehlerhafter Name oder der Name einer bereits auf der Diskette vorhandenen Kartei eingegeben, so nimmt das Programm diesen Namen nicht an (eine einmal eingerichtete Kartei kann aus Sicherheitsgründen nur im Direktmodus gelöscht werden!).

Z. 2850:
Wird die erste Karte (die Identifikationskarte) überschrieben, werden die Sortierroutinen übersprungen.

Z. 3100:
Die Zählvariablen aus der Texteingaberoutine werden hier

um 1 zurückgesetzt, damit das "*" nicht versehentlich bei der weiteren Texteingabe gelöscht werden kann. Anderenfalls würde das Programm die Eingabe als ganz "normale" Eingabe ansehen und entsprechend einordnen. Die Länge der Titelzeile verkürzt sich dadurch um 1.
Z. 3510:
Die Tabelle der Stichwörter umfaßt max. 108 verschiedene Einträge. Damit paßt sie genau in die jeweilige Bildschirmmaske.

Erläuterungen zum Sortierprogramm LITSORT:
Mit dem Zusatzprogramm LITSORT.BAS (Listing 2) kann man die Stichwörterkartei (*.STI), die mit jeder Karteikarten-Kartei (*.KRT) gekoppelt ist, auf den aktuellen Stand bringen, wenn man in der Textkartei häufig Karteneinträge geändert hat.
Wie bereits in der Beschreibung des LITBOX-Programms erwähnt, dient die Stichwörterkartei (*.STI) nur dazu, sämtliche bei der Eingabefunktion als "Stichwörter" eingegebenen Begriffe alphabetisch geordnet aufzulisten. Sie hat also mit der eigentlichen Suchfunktion des Programms nichts zu tun, sondern kann sowohl bei der Eingabe als auch bei der Suche als "Gedächtnisstütze" hilfreich sein: Welche Stichwörter habe ich schon verwendet bzw. nach welchen Begriffen könnte ich denn noch suchen?

```
HINWEISE ZUM KOPIEREN VON LITERATURKARTEIEN UND ZUM START DES LITBOX-PROGRAMMS

==*==) Zum Kopieren einer Literaturkartei (Nr. 1?? oder 2??) vom Laufwerk A auf die
       RAM-Disk. (M) die entsprechende KARTEI-Diskette in das Laufwerk schieben und

                [EXTRA] und [1]  bzw.  [EXTRA] und [2]  drücken

==*==) Zum Kopieren der Literaturkartei von der RAM-Disk (M) auf das Laufwerk A
       die entsprechende Ziel-Diskette in das Laufwerk schieben und

                          [EXTRA] und [*] drücken

==*==) und/oder

       zum Starten des Literaturkartei-Programms die START-Diskette einschieben und

                               [##] drücken
```

Datei HINWEISE.LIT *Bild 7*

Falls man beim Ausfüllen neuer Karteikarten auf insgesamt über 108 verschiedene Stichwörter kommt, hat das lediglich Auswirkungen auf diese "Gedächtnisstütze": durch jedes neue Stichwort wird das bis dahin letzte aus der Stichwörterkartei "rausgeworfen" und erscheint folglich beim Drücken des '?' nicht mehr auf dem Bildschirm. In der eigentlichen Textkartei (*.KRT) wird es selbstverständlich eingetragen.
Wird ein Stichwort bei der Eingabe einer neuen Karte versehentlich falsch geschrieben und gespeichert, so kann es zwar leicht in der Textkarte (mit '*') verbessert werden, bleibt dann aber in der fehlerhaften Form (und der korrigierten!) in der Stichwörterkartei stehen. Dies ist zunächst nicht weiter schlimm, da es, wie gesagt, die Suchfunktion nicht beeinträchtigt. Erst, wenn das häufiger vorkommt, oder wenn noch mehrere Stichwörter in der Liste stehen, deren zugehörige Textkarten längst alle wieder neu beschrieben wurden, dürfte es ratsam sein, die Stichwörterkartei mit Hilfe des Sortier-Programms LITSORT.BAS neu zu erstellen.
Aus diesem Grund ist das Programm auch recht einfach gehalten und verzichtet auf eine aufwendige Benutzerführung. Wer seine Stichwörterkartei damit auf den neuesten Stand bringen möchte, muß lediglich das Basic und dieses Programm laden, die Diskette mit der zu sortierenden Kartei einlegen, das Programm starten und dann den Karteinamen eintippen und etwas Geduld haben (das Heraussuchen und Sortieren der Stichwörter dauert bei einer voll-

```
1810 x=VAL(taste$)'-----> Nr. der einzeln zu druck
enden Karte
1820 IF karte$(x)="" THEN 1750'-----> wdh., da fal
sche Eingabe
1830 '------------------ eine einzelne Karteikarte
 ausdrucken ------------------------------
1840 LPRINT LEFT$(karte$(x),55);"*";MID$(karte$(x)
,56,14);" ";MID$(karte$(x),70,14)
1850 LPRINT MID$(karte$(x),84,10);"*";MID$(karte$(
x),94,75)
1860 LPRINT
1870 GOTO 1750'----->>> nächsten Befehl abfragen (
Drucken oder weiter)
1880 '---------------------- alle Karteikarten aus
drucken ---------------------------------
1890 FOR x=1 TO 7
1900 IF karte$(x)="" THEN 1960
1910 LPRINT LEFT$(karte$(x),55);"*";MID$(karte$(x)
,56,14);" ";MID$(karte$(x),70,14)
1920 LPRINT MID$(karte$(x),84,10);"*";MID$(karte$(
x),94,75)
1930 LPRINT
1940 NEXT
1950 GOTO 1750'----->>> nächsten Befehl abfragen (
Drucken oder weiter)
1960 PRINT FNpkt$(5,90,"");cl.u$;c.an$:nr=1:RETURN
'-----> UP Drucken verlassen
1970 '============== HP Leere Kartei erstellen (E
inspr. aus 730) ======================
1980 kartei$=""
1990 PRINT FNpkt$(20,23,cl.u$+"Wie soll die neu zu
 erstellende KARTEI heißen?")
2000 PRINT FNpkt$(22,7,"(Erstes Zeichen eine 1 ode
r 2): "+inv.an$+SPACE$(9)+".KRT "+inv.aus$)
2010 PRINT FNpkt$(24,20,"Im Karteinamen sind folge
nde Zeichen nicht erlaubt:")
2020 PRINT FNpkt$(26,27,inv.an$+" ! $ & / ( ) ? Ü
+ * ö Ä , ; . : - < > "+inv.aus$)
2030 PRINT FNpkt$(22,40,""+inv.an$);:laenge.wort=8
'-----> Cursorpos./Wortlänge festlegen
2040 GOSUB 3290'----->>> UP Texteingabe
2050 PRINT inv.aus$
2060 IF zchn$<>CHR$(8) THEN 2070 ELSE 470'----->>>
 weiter oder Menü 1
2070 IF LEFT$(wort$,1)="1" OR LEFT$(wort$,1)="2" T
HEN 2080 ELSE 2150
2080 FOR x=1 TO 8'-----> Leerstellen aus dem Karte
inamen entfernen
2090 zchn.kart$=MID$(wort$,x,1)
2100 IF zchn.kart$<>" " THEN kartei$=kartei$+zchn.
kart$
2110 NEXT
2120 stichwort$=kartei$+".STI"'-----> Name der Sti
chwortkartei
2130 kartei$=kartei$+".KRT"'-----> Name der Datenk
artei
2140 GOTO 2170
2150 PRINT FNpkt$(18,10,">>> "+inv.an$+" Die Einga
be war fehlerhaft bzw. den Karteinamen gibt es sch
on "+inv.aus$+" <<<"):GOTO 1980'----->>> Wiederhol
en
2160 RESUME 2150
2170 ON ERROR GOTO 2160
2180 IF FIND$(kartei$)<>"" THEN 2150'-----> Prüfen
 des eingegebenen Karteinamens
2190 PRINT FNpkt$(24,10,cl.z$+"Hier können noch nä
here Angaben zum Inhalt der Kartei gemacht werden:
")
2200 PRINT FNpkt$(26,3,cl.z$+inv.an$+SPACE$(84)+in
v.aus$)
2210 PRINT FNpkt$(26,4,""+inv.an$+c.an$);:laenge.w
ort=83'----> Cursorpos./Wortlänge
2220 GOSUB 3290'----->>> UP Texteingabe
2230 IF zchn$<>CHR$(8) THEN thema$=wort$ ELSE 470'
-----> weiter oder Menü 1
2240 OPEN "R",1,kartei$:OPEN "R",2,stichwort$,14'-
---->Die neue Datei öffnen
2250 FIELD 1,128 AS d.karte$:FIELD 2,14 AS d.stich
wort$
2260 PRINT inv.aus$;cl$;FNpkt$(21,12,"Auf der Disk
ette werden 600 leere Karteikarten angelegt,")
2270 PRINT FNpkt$(23,12,"außerdem 108 Karten für S
tichwörter...")
2280 LSET d.karte$=thema$:MID$(d.karte$,95,1)="2"'
-----> Thema der Kartei und Nr. der 2. Karte in di
e 1. Karte eintragen
2290 PUT 1,1
2300 LSET d.karte$=CHR$(0)
2310 '-------------------- leere Textkartei (Datei
```

```
) anlegen ------------------------------
2320 FOR x=1 TO 600
2330 PUT 1
2340 NEXT
2350 '---------------- leere Stichwörterkartei (Da
tei) anlegen) --------------------------
2360 LSET d.stichwort$=STRING$(14,127)
2370 FOR x=1 TO 108
2380 PUT 2
2390 NEXT
2400 PRINT FNpkt$(26,28,inv.an$+" Die Kartei ist e
ingerichtet "+inv.aus$)
2410 FOR x=1 TO 500:NEXT:CLOSE:GOTO 470'-----> "Wa
rteschleife"/Rücksprung zum Menü 1
2420 '============ HP Schreiben neuer Karteikarten
 (Einspr. aus 1120) ===================
2430 FIELD 1, 55 AS d.titel$, 14 AS d.stichwort1$,
 14 AS d.stichwort2$, 10 AS d.stichwort3$,35 AS d.
quelle$
2440 '------------------ Informationen zur Program
msteuerung ----------------------------
2450 PRINT cl$;c.an$
2460 PRINT FNpkt$(13,4,inv.an$+" ^ "+inv.aus$+" (i
n den Text eingefügt) => Wiederholung der gesamten
 Eingabe")
2470 PRINT FNpkt$(15,4,inv.an$+" ? "+inv.aus$+" (w
ährend der Stichworteingabe) => Auflistung der ges
peicherten Stichwörter")
2480 PRINT FNpkt$(17,0,">>> Soll eine bereits ausg
efüllte Karte überschrieben werden, so muß am Anfa
ng der")
2490 PRINT FNpkt$(18,4,"Autor/Titel-Zeile ein "+in
v.an$+" * "+inv.aus$+" eingegeben werden. Außerdem
 muß die "+inv.an$+" Reg.Nr. "+inv.aus$+" der zu")
2500 PRINT FNpkt$(19,4,"löschenden Karte bekannt s
ein")
2510 GET 1,1
2520 nr.satz=VAL(d.quelle$):krt.inhalt$=d.titel$'-
--> 1. leere Karte/Info. üb. d. Kartei
2530 '-------------------- Bildschirmmaske für Tex
teingabe ------------------------------
2540 PRINT FNpkt$(0,0,inv.an$+" Ergänzen: "+inv.au
s$+" "+kartei$+" "+inv.an$+"  Bisherige Einträge i
```

geschriebenen Kartei (=600 Karteikarten) ca. 10 Minuten!).

Zusammenstellung einer Startdiskette für das Literaturprogramm:

Um das Literaturkartei-Programm ohne größere Umstände starten zu können, empfiehlt es sich, dafür eine gesonderte Start-Diskette zusammenzustellen. Dazu müssen folgende Programme von der System-Diskette auf eine Arbeitsdiskette kopiert werden (s. Handbuch, Kapitel 3.2 bzw. Startdisketten anders erstellt):

J???CPM3.EMS, BASIC.COM, PALETTE.COM, PIP.COM, SETDEF.COM, SETKEYS.COM, SUBMIT.COM

Ferner erstellt man mit Hilfe des auf der System-Diskette vorhandenen Editors RPED.BAS (Aufruf durch **basic rped**) die in den Bildern 1 - 7 als Hardcopys gezeigten .SUB-Programme auf der Arbeitsdiskette. Außerdem müssen sich die zwei Basic-Programme LITBOX.BAS und LITSORT.BAS auf der Diskette befinden.

Falls man sich beim Abtippen der Hardcopys nicht irgendwo vertan hat, müßte es klappen: Nach dem Einschalten des Joyce schiebt man die LITBOX-Startdiskette in das Laufwerk und beobachtet, wie nach Laden von CP/M zunächst der Bildschirm umgeschaltet und die Suchfolge für die Laufwerke bestimmt wird. Danach werden die Tasten "benutzerfreundlich" belegt (mehr dazu voraussichtlich im nächsten Heft) und alle für den weiteren Programmablauf notwendigen Hilfsprogramme in die RAM-Disk kopiert. Anschließend wird auf dem Bildschirm erläutert, wie man die Karteien durch einfachen Tastendruck von Laufwerk A: nach M: und wieder zurück kopiert bekommt, bzw. das LITBOX-Programm ebenfalls durch einen Tastendruck gestartet wird.

(Klaus Wiltsch/MC)

```
n der Kartei:  "+STR$(nr.satz-1)+" "+inv.aus$)
2550 PRINT FNpkt$(2,0,"Autor/Titel:  "+inv.an$+SPA
CE$(57)+inv.aus$);" (55 Zchn.)"
2560 PRINT FNpkt$(4,0,"1. Stichwort: "+inv.an$+SPA
CE$(16)+inv.aus$+" (14 Zchn.)")
2570 PRINT FNpkt$(6,0,"2. Stichwort: "+inv.an$+SPA
CE$(16)+inv.aus$+" (14 Zchn.)")
2580 PRINT FNpkt$(8,0,"Materialart:  "+inv.an$+SPA
CE$(12)+inv.aus$);" (10 Zchn.)"
2590 PRINT FNpkt$(10,0,"Quelle:        "+inv.an$+SP
ACE$(37)+inv.aus$);" (35 Zchn.)"
2600 PRINT FNpkt$(11,0,STRING$(89,138))
2610 '-------------- Hinweis, wenn die Kartei voll
geschrieben ist -------------------------
2620 IF nr.satz>600 THEN 2630 ELSE 2650
2630 PRINT FNpkt$(25,8,CHR$(7)+inv.an$+" >>> Die K
artei ist voll. Es können nur noch Karten geändert
 werden! <<< "+inv.aus$+CHR$(7))
2640 '------------------------------ Texteingabe -
--------------------------------------------
2650 PRINT FNpkt$(2,15,""+inv.an$);:laenge.wort=55
'----> Cursorpos./Wortlänge
2660 GOSUB 3290'----->>> UP Texteingabe
2670 IF INSTR(wort$,"^") THEN 2530'----->>> Wieder
holen
2680 IF zchn$<>CHR$(8) THEN titel$=wort$ ELSE 980'
------>>> Menü 2
2690 PRINT FNpkt$(4,15,"");:laenge.wort=14'----->
Cursorpos./Wortlänge
2700 GOSUB 3290'----->>> UP Texteingabe
2710 IF INSTR(wort$,"^") THEN 2530'----->>> Wieder
holen
2720 IF zchn$<>CHR$(8) THEN stichwort1$=wort$ ELSE
 980'----->>>Menü2
2730 PRINT FNpkt$(6,15,"");:laenge.wort=14'----->
Cursorpos./Wortlänge
2740 GOSUB 3290'----->>> UP Texteingabe
2750 IF INSTR(wort$,"^") THEN 2530'----->>> Wieder
holen
2760 IF zchn$<>CHR$(8) THEN stichwort2$=wort$ ELSE
 980'----->>> Menü2
2770 PRINT FNpkt$(8,15,"");:laenge.wort=10'----->
Cursorpos./Wortlänge
2780 GOSUB 3290'----->>> UP Texteingabe
2790 IF INSTR(wort$,"^") THEN 2530'----->>> Wieder
holen
2800 IF zchn$<>CHR$(8) THEN stichwort3$=wort$ ELSE
 980'----->>> Menü2
2810 PRINT FNpkt$(10,15,"");:laenge.wort=35'----->
 Cursorpos./Wortlänge
2820 GOSUB 3290:PRINT inv.aus$'----->>> UP Textein
gabe
2830 IF INSTR(wort$,"^") THEN 2530'----->>> Wieder
holen
2840 IF zchn$<>CHR$(8) THEN quelle$=wort$ ELSE 980
'------>>> Menü2
2850 IF nr.loe=1 THEN 2900
2860 IF stichwort1$="" THEN 2880
2870 LSET sw$(109)=stichwort1$:GOSUB 3210'----->>>
 Sortierroutine
2880 IF stichwort2$="" THEN 2900
2890 LSET sw$(109)=stichwort2$:GOSUB 3210'----->>>
 Sortierroutine
2900 LSET d.titel$=titel$
2910 LSET d.stichwort1$=stichwort1$
2920 LSET d.stichwort2$=stichwort2$
2930 LSET d.stichwort3$=stichwort3$
2940 LSET d.quelle$=quelle$
2950 sw.neu$="ja"
2960 IF LEFT$(titel$,1)="*" THEN 3060
2970 IF nr.satz>600 THEN 2510'----->>> Wdh., da Ka
rtei bereits vollgeschrieben ist
2980 '------------------- Text in Karteikarte (Date
i) schreiben ---------------------------
2990 PUT 1,nr.satz
3000 nr.satz=nr.satz+1
3010 '------------ Einträge 1. Karteikarte (1. Dat
ensatz) aktualisieren ------------------
3020 LSET d.titel$=krt.inhalt$:LSET d.stichwort1$=
"":LSET d.stichwort2$="":LSET d.stichwort3$="":LSE
T d.quelle$=STR$(nr.satz)
3030 PUT 1,1'-----> Nr. der (neuen) nächsten leere
n Karteikarte eintragen
3040 GOTO 2510'----->>> nächste Karte ausfüllen
3050 '------------ Karteikarte überschreiben/lösch
en (Einspr. aus 2880) -------------------
3060 LSET d.titel$=MID$(titel$,2,54)'-----> "*" au
s dem Titel entfernen
3070 PUT 1,nr.loe
3080 nr.loe=0:PRINT FNpkt$(11,90,cl.u$):GOTO 2510'
----->>> nächste Karte ausfüllen
3090 '-------- UP Reg.Nr. der zu löschenden Karte
eingeben (Einspr. aus 3380) ------------
3100 PRINT CHR$(27);"j";:nr.zchn=0:laenge.wort=lae
nge.wort-1'-----> Cu.pos. speichern/Zähler zurücks
etzen
3110 PRINT FNpkt$(12,0,cl.u$):PRINT" Welche Kartei
karte soll überschrieben/gelöscht werden? "
3120 PRINT FNpkt$(15,25,cl.z$+" Reg.Nr.:"+inv.aus$
);" ";:LINE INPUT; nr.loe$:PRINT inv.an$;
3130 IF nr.loe$<>"" THEN nr.loe=VAL(nr.loe$) ELSE
2450'-----> Weiter od. neue Eingabe
3140 IF nr.loe<nr.satz THEN 3160
3150 PRINT FNpkt$(17,0," Die eingegebene Nr. ist g
rößer als die Zahl der bisherigen Einträge! "):GOT
O 3120
3160 GET 1,nr.loe
3170 karte.loe$=d.karte$+STRING$(25,32)+"Reg.Nr.:
"+STR$(nr.loe)'-----> Karte mit Reg.Nr.versehen
3180 PRINT FNpkt$(17,0,LEFT$(karte.loe$,55)+CHR$(1
33)+MID$(karte.loe$,56,14)+" "+MID$(karte.loe$,70,
14))'----> zu löschende Karte am Bildschirm ausgeb
en
3190 PRINT FNpkt$(18,0,MID$(karte.loe$,84,10)+MID$
(karte.loe$,94,75));
3200 PRINT CHR$(27);"k";:RETURN'-----> gespeichert
e Cursorposition
3210 '------ UP Stichwört. alphabetisch einsortier
en (Einspr. aus 2730 u. 2750) ----------
3220 FOR x=1 TO 108'-----> Prüfen, ob das Stichwor
t bereits vorkommt
3230 IF UPPER$(sw$(x))=UPPER$(sw$(109)) THEN RETUR
N
3240 NEXT
3250 FOR x=109 TO 2 STEP -1'-----> Einsortieren de
s Stichwortes
3260 IF UPPER$(sw$(x))<UPPER$(sw$(x-1)) THEN SWAP
sw$(x),sw$(x-1)
3270 NEXT
3280 RETURN
3290 '========= UP Eingaberoutine für Texte (Einsp
r. aus diversen Zeilen) ================
3300 nr.zchn=1
3310 wort$=""
3320 PRINT c.an$;
3330 zchn$=INKEY$:IF zchn$="" THEN 3330
3340 IF zchn$=CHR$(8) OR zchn$=CHR$(13) THEN RETUR
N'--->>> Abbruch bzw. Forts. d. Progr's.
3350 IF zchn$="?" AND laenge.wort=14 THEN GOSUB 35
20:GOTO 3330'-----> Tab. der Stichwörter ausdrucke
n, dann weiter
3360 IF zchn$=CHR$(27) THEN 3640
3370 zchn=ASC(zchn$)
3380 IF zchn=127 AND nr.zchn=1 THEN 3330
3390 IF zchn<>127 AND nr.zchn>laenge.wort THEN 333
0
3400 IF zchn=127 THEN 3410 ELSE 3450
3410 PRINT c.li$;:PRINT CHR$(32);:PRINT c.li$;
3420 wort$=LEFT$(wort$,LEN(wort$)-1)
3430 nr.zchn=nr.zchn-1
3440 GOTO 3330'----->>> nächstes Zeichen abfragen
3450 IF zchn>31 AND zchn<127 THEN 3460 ELSE 3330
3460 PRINT CHR$(zchn);
3470 wort$=wort$+CHR$(zchn)
3480 IF nr.zchn=1 AND laenge.wort=55 AND zchn$="*"
 THEN GOSUB 3100
3490 nr.zchn=nr.zchn+1
3500 GOTO 3330'----->>> nächstes Zeichen abfragen
3510 '========= UP Ausgabe der Tabelle der Stichw
örter (Einspr. aus 3350) ==============
3520 PRINT CHR$(27);"j":PRINT FNpkt$(11,90,""+cl.u
$)'---> Cu.pos. speichern/neue Cu.pos.
3530 x=1:tb=0'-----> Zählvariable und Tabulatorvar
iable zurücksetzen
3540 FOR y=12 TO 29
3550 IF ASC(sw$(x))=127 THEN 3610'-----> letztes S
tichwort erreicht
3560 PRINT FNpkt$(y,tb,sw$(x))
3570 x=x+1
3580 IF x>108 THEN 3610
3590 NEXT y
3600 tb=tb+15:GOTO 3540
3610 PRINT CHR$(27);"k";:RETURN'-----> gespeichert
e Cursorposition
3620 '====== UP Programmablauf zurücksetzen/beende
n (Einspr. aus diversen Zeilen) ======
3630 PRINT cl$;inv.aus$;c.an$:CLOSE:RETURN
3640 GOSUB 3650:GOTO 3730
```

```
3650 IF sw.neu$<>"ja" THEN RETURN'-----> prüfen, o
b Eingaben erfolgten
3660 '------------ Tabelle der Stichwörter in die
*.STI-Datei übertragen ----------------
3670 FOR x=1 TO 108
3680 IF ASC(sw$(x))=127 THEN RETURN
3690 LSET d.stichwort$=sw$(x)
3700 PUT 2,x
3710 NEXT
3720 RETURN
3730 PRINT cl$;inv.aus$;c.an$
3740 PRINT FNpkt$(4,30,inv.an$+SPACE$(32))
3750 PRINT FNpkt$(5,30," ENDE DES KARTEIKARTENPROG
RAMMS ")
3760 PRINT FNpkt$(6,30,SPACE$(32)+inv.aus$)
3770 PRINT FNpkt$(10,10,"Im Moment befinden sich f
olgende LITERATURKARTEIEN in den Laufwerken:")
3780 OPTION FILES "M"
3790 PRINT FNpkt$(12,32,"Laufwerk M: "+inv.an$);
3800 IF FIND$("*.KRT")="" THEN 3820
3810 FILES("*.KRT")
3820 OPTION FILES "A"
3830 PRINT FNpkt$(14,32,inv.aus$+"Laufwerk A: "+in
v.an$);
3840 IF FIND$("*.KRT")="" THEN 3860
3850 FILES("*.KRT")
3860 PRINT FNpkt$(17,18,inv.aus$+"Falls sich eine
"+inv.an$+" geänderte "+inv.aus$+" Kartei im Laufw
erk M befindet, ")
3870 PRINT FNpkt$(18,17," nicht vergessen, sie wie
der auf die Diskette zu kopieren!  "+inv.aus$)
3880 PRINT FNpkt$(21,0,"Dazu zunächst durch Drücke
n der Tasten "+inv.an$+" ALT + SHIFT + ## "+inv.au
s$+" in den CP/M-Modus zurück,")
3890 PRINT FNpkt$(23,0,"dann durch Drücken der "+i
nv.an$+" COPY "+inv.aus$+"-Taste die Hinweise für
die Kopierprogramme aufrufen.")
3900 PRINT FNpkt$(25,0,"");:END
```

Listing 2:
Sortierprogramm LISTSORT.BAS

```
10 DIM sw$(110)
15 PRINT:PRINT:FILES:PRINT:PRINT
20 PRINT"Welche Kartei soll sortiert werden?"
30 PRINT"Nur den Dateinamen, nicht den Dateityp (.
KRT bzw. .STI) eingeben!"
40 LINE INPUT kartei$
50 PRINT"Jetzt wird ganz wild sortiert..."
60 PRINT"Eventuell reicht die Zeit für einen Kaffe
e!-"
70 kartei.krt$=kartei$+".krt":kartei.sti$=kartei$+
".sti"
80 FOR x=1 TO 109:sw$(x)=STRING$(14,127):NEXT'----
-> "leere" Tabelle anlegen
90 sw$(0)=STRING$(14,32)'----> niederwertigstes Wo
rt beim Vergleichen
100 OPEN "R",1,kartei.krt$:OPEN "R",2,kartei.sti$,
14
110 FIELD 1, 55 AS d.titel$, 14 AS d.stichwort1$,
14 AS d.stichwort2$, 10 AS d.stichwort3$,35 AS d.q
uelle$
120 FIELD 2, 14 AS d.stichwort$
130 GET 1,1
140 nr.letzte=VAL(d.quelle$)'----> Nr. der letzten
 beschriebenen Karteikarte
150 FOR x=2 TO nr.letzte
160 GET 1,x
170 wrt$=d.stichwort1$
180 GOSUB 230'----->>> Sortierroutine
190 wrt$=d.stichwort2$
200 GOSUB 230'----->>> Sortierroutine
210 NEXT
220 GOTO 310'----->>> neue Liste übertragen
230 sw$(109)=wrt$
240 FOR y=0 TO 108
250 IF UPPER$(sw$(y))=UPPER$(sw$(109)) THEN RETURN
260 NEXT
270 FOR y=109 TO 2 STEP -1
280 IF UPPER$(sw$(y))<UPPER$(sw$(y-1)) THEN SWAP s
w$(y),sw$(y-1)
290 NEXT
300 RETURN
310 FOR X=1 TO 108'-----> die neu zusammengestellt
e Liste in die Datei übertragen
320 LSET d.stichwort$=sw$(x)
330 PUT 2,x
340 NEXT
350 PRINT CHR$(7);CHR$(7);CHR$(7)
360 PRINT"Fertig!"
370 END
```

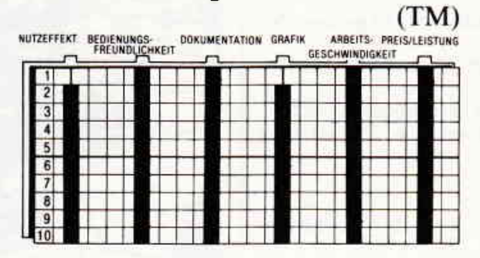
STAR TEXTER

Hersteller: Sybex
Vertrieb: Sybex
Steuerung:Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: 85,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Ob man nun bei Textverarbeitungen auf Komfortabilität oder Unmengen Speicherplatz besteht, bleibt jedem selbst überlassen.
Textverarbeitungen, die beides bieten, gibt es unter CP/M in rauhen Mengen. Für Leute, die sich mit dem Betriebssystem jedoch nicht anfreunden können, gab es bisher nur die eine oder andere Alternative. Neu auf dem Markt ist der Star Texter der Firma Sybex. Hier findet der Heimanwender alles, was er benötigt, in einem Programm. Selbst mir, der ja nun schon so manches Professionelle gesehen hat, fiel im ersten Moment nichts mehr ein. Einfachheit der Handhabung und Leistungsdaten paaren sich hier so perfekt, daß eine Verbesserung kaum noch möglich ist.
Angefangen vom hervorragenden Druckerinstallationsmenue, das es erlaubt, Star Texter an jedem, wirklich jedem Drucker zu betreiben, bis hin zu Funktionen wie Mehrspaltensatz, hat der Star Texter alle Einrichtungen, die ein professionelles Programm ausmachen. Diese Textverarbeitung wird hier auch nur deshalb nicht als professionell bezeichnet, weil die Handhabung wieder viel zu einfach ist.
Star Texter ist für mich nicht unbedingt das, was man als Anwenderprogramm bezeichnen kann. Es geht einen völlig neuen Weg, der am ehesten noch dem eines Kurses mit beigelieferter Software entspricht. Diese Software hat es allerdings in sich.
Nicht nur, daß sowieso alles integriert ist, was standardmäßig in eine Textverarbeitung gehört, sondern auch noch Features wie Grafik und Text mischbar, Layoutanzeige und mehrere Tastaturen – im Fachjargon Firlefanz genannt – aber für den Heimanwender Optionen, auf die er in keinem Fall verzichten sollte.
Der Kurs (Pardon: das Handbuch) ist sehr gut geschrieben, und nach der Lektüre dieses Buches sollte auch ein sabbernder Steinzeitmensch keine Schwierigkeiten mehr im Umgang mit Star Texter haben.
Programmierung in M-Code ist bei Textverarbeitungen dieser Preisklasse zwar nicht unbedingt üblich, aber wenn Sybex sich das leisten kann – bitte sehr. Dem Anwender kommt es auf jeden Fall zugute. Geschwindigkeit zählt bei Star Texter zu den absoluten Trümpfen. Selten habe ich mit einer professionellen Textverarbeitung so flink gearbeitet, wie mit diesem ausgeklügelten System.
Last not least hat jedoch auch dieses Programm seine Macken. Ein BUG, der mich sehr störte, ist, daß das Programm von sich aus entscheidet, wann ein Text löschenswert ist oder nicht. Das heißt nicht, daß Texte irgendwann mal während des Schreibens in der Memory verschwinden und von dort nie zurückkehren, sondern Star Texter weigert sich manchmal beharrlich, den Anweisungen seines Users zu folgen und löscht Texte nicht!
Dies kann von Optimisten als Vorstufe zur künstlichen Intelligenz angesehen werden, für einen Kritiker ist es ein BUG.
Jedem, der den Einstieg in die Textverarbeitung erlernen möchte, ist das Paket wärmstens zu empfehlen. Ein späteres Umsteigen auf größere Systeme ist ja immer noch möglich. Star Texter leistet jedoch die ganze Vorarbeit und vermittelt das Grundwissen, das man zur Handhabung eines solchen Systems braucht, auf optimale Weise.
Leichte Fehler im Programm, die noch verbessert werden könnten, nimmt man da gerne mal in Kauf.

(TM)



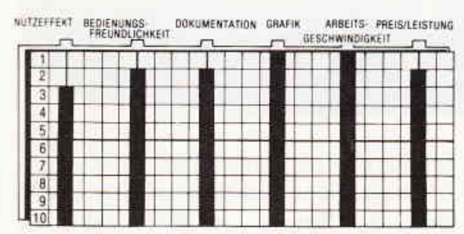
MOS

Hersteller: Star Division
Vertrieb: Star Division
Autor: A. Zallmann
Steuerung: Joystick/Maus
Monitor: Farbe/Grün
Programm: 100% M-Code
Preis: 198,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

MOS (Mouse Operating System ist eine Verballhornung von DOS (Disk Operating System), die sehr gut den Kern der Sache trifft. Anders als Operations-System kann man dieses Programm nicht beschreiben.
Ähnlich, wie z.B. bei den Computern Atari oder Amiga, wird dem Schneider-Anwender hier eine Benutzeroberfläche bereitgestellt. Diese Benutzeroberfläche läßt sich von Basic aus über neue Befehle programmieren. Wer jetzt stöhnt, "die 2789ste RSX-Erweiterung", der hat sich geirrt. MOS benötigt die RSX-Balken nicht. Alle Eingaben werden direkt vom Locomotive-Basic als neue Befehle akzeptiert. So etwas geht natürlich nur in M-Code, was jedem auch klar wird, wenn er sich die Geschwindigkeit des Systems betrachtet. Bis zu 20 echte Overlay-Windows können erzeugt werden. Diese können sich überlappen, ohne sich dabei gegenseitig die Inhalte zu zerstören. Auch bei einem Scroll bleiben die Fenster noch erhalten. Durch einfache Angabe von Parametern lassen sich Closepointer, Bars und Überschriften festlegen. Wem diese Begriffe nichts sagen, der kann getrost dem enormen Handbuch vertrauen. Spätestens nach dessen Lektüre sind sie keine Fremdworte mehr.
Lange war dieses MOS angekündigt, und so mancher Anwender fragte sich schon, ob das alles vielleicht nur einem wilden Gerücht entstammte. Aber nun steht das System, und scheint eine kleine Sensation auf dem CPC zu werden. Realisten werden sich natürlich fragen, wo eigentlich die Anwendung einer solchen Erweiterung liegen soll. Tatsächlich ist das Einsatzgebiet der Software recht dürftig. Außer für private Anwendungen kann man es ja aufgrund des Copyrights nicht verwenden. Zum Paket gehört jedoch auch noch eine entsprechende Maus, und schon hat man eine Anwendung. Die Maus entspricht der Amigamaus des Commodore und wird allen professionellen Ansprüchen gerecht.
Die Programmierung mit Pull-Down-Windows bereitet sehr viel Spaß, und man ertappt sich ziemlich schnell dabei, daß man alle seine Programme auf einmal mit umfangreichen Menues versieht, nur um immer wieder das Vergnügen des Anklickens zu haben.
GEM und ähnliche Betriebssysteme laufen ja auch auf jedem größeren PC, und niemand sagt, daß so etwas eine Spielerei ist. (Außer Firmen, die so etwas nicht haben.)
MOS ist ein Tool, dessen Anwendung alleine für Heimprogrammierer bestimmt ist. Diese werden allerdings sehr viel Spaß an dem Programm haben. Anwender, die sich sowieso eine Maus zulegen wollten, kommen an MOS auch nicht vorbei. Ein wenig leid tun mir die, die schon einen sol-

chen Nager besitzen und die Software nicht einzeln erwerben können. Auch das Fehlen eines Grafikprogramms ist sehr bedauerlich. Sicherlich werden jedoch noch weitere Programme, die unter MOS laufen, auf dem Markt erscheinen.
Völlig neue Möglichkeiten werden den CPC-Anwendern durch MOS nicht eröffnet, aber bestehende Features wurden wesentlich verbessert und durch mächtige Befehle der Programmieraufwand auf ein Minimum begrenzt. (TM)

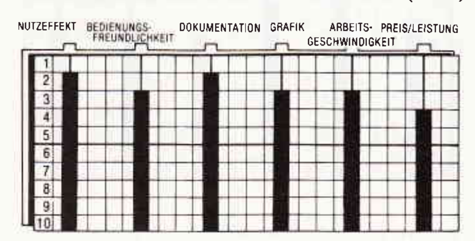


Profi Basic

Hersteller: Power Soft
Vertrieb: Gepo Soft
Steuerung: Tastatur
Monitor: Farbe/Grün
Programm: 100% M-Code
Preis: 159,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Immer besser und immer mächtiger werden die Basic-Erweiterungen der CPC Rechner. Wehe dem geplagten Anwender, der sich einmal für eine entscheidet. Spätestens nach einem Monat kommt eine Neue auf den Markt, die mit Sicherheit die "allerletzte" Entwicklung vom Vormonat noch bei weitem übertrumpft.
Diesmal ist es Gepo Soft, die mit einer leistungsstarken Basic-Erweiterung aus Frankreich auf den Markt kommt. Und tatsächlich, PROFI BASIC verdient seinen Namen zu recht.
Starke Befehle für Grafik und Sound stehen im Vordergrund dieses Tools, und lassen es in dem Licht erscheinen, daß es direkt für die Programmierer von Spielen entwickelt wurde. Aber auch der an Anwendungen Interessierte findet hier interessante Features, die bisher noch nicht da waren.
Komplexe Steuerungen von Windows animieren dazu, seine Privatanwendung durch Pull-Down-Windows zu verschönen. Die Befehle für 3D-Grafiken könnten manchem privaten CAD-Programm den letzten Schliff geben.
Hauptvorteil des Profi Basic gegenüber anderen Basic-Erweiterungen ist, meiner Meinung nach, die Festlegung auf ein Thema. Durch die Geschwindigkeit und Komfortabilität des Tools ist eine Verbesserung so schnell nicht möglich. Der an Grafik Interessierte kann hier zunächst einmal davon ausgehen, daß er in den nächsten drei(?) Monaten von neuen Erweiterungen verschont bleibt.
Somit ist zumindest ein Kennenlernen der über 80 neuen Befehle gesichert. Wer hat schon Lust, jeden Monat umzusteigen.
Ein ausgezeichnetes Handbuch, das in die Grundlagen der Grafikprogrammierung einführt, und starke Befehle wie AXES, ZOOM, ROTATE, ZONE und Hardcopys machen das Arbeiten, zumindestens mit der Grafik, zum Kinderspiel.
Ich bin gespannt darauf, wie die Konkurrenz es versuchen wird, dieses Programm in der Leistungsfähigkeit noch zu überbieten. Mir fällt zu diesem Thema schon lange nichts mehr ein, und ich sitze jedesmal sehr erstaunt vor meinem CPC und frage mich, zu welchen Wunderdingen dieser kleine Kasten noch fähig ist.
Vielleicht hat ja die Erweiterungs-Sensation des nächsten Monats Befehle wie PACMAN oder DEFENDER. Damit wäre der Programmierer dann jeder eigenen Geistesleistung völlig enthoben, und noch besser gehts doch wirklich nicht – ODER? (TM)

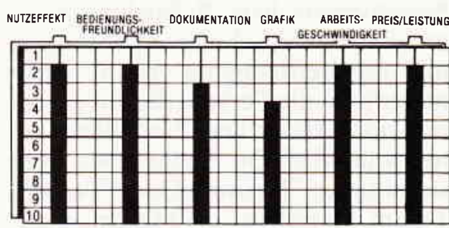


Telecom 1000

Hersteller: RSE
Vertrieb: RSE
Steuerung: Tastatur
Monitor: Farbe/Grün
Programm: Basic/M-Code
Preis: 79,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

DFÜ-Anwender mit der original Schneider-Schnittstelle haben wieder einen Grund zur Freude. Die nach dem Erwerb der RS-232-Schnittstelle ein wenig stiefmütterlich behandelten User bekommen mit Telecom 1000 ein Kommunikationspaket der obersten Spitzenklasse in die Hand. Kaum zu glauben, daß der CPC, der sich bis vor kurzem noch über provisorische RS-232-Lösungen, die es für nahezu jeden Steckplatz der Geräterückseite gab, mit seiner Umwelt verständigen konnte, sich jetzt als echtes Datenterminal hervortut.
Die Entwicklung hierzu hat bei einem Z80 Rechner schon recht lange gedauert. Aber die Kombination vernünftiger Hardware, mit jetzt auch der entsprechenden Software, lassen diese Zeiten schnell vergessen.
Telecom 100 entspricht genau dem, was bei der Entwicklung der RS-232 vergessen wurde. Ein leistungsfähiger Treiber der Schnittstelle für Hacker, Hobbybastler und andere DFÜ-Freaks.
Integrierter Texteditor und ein Speicher von 32 KB sind Leistungsdaten, mit denen Apple- und IBM-Besitzer seit Jahren umgehen. Menueführung und ein großer Parametereditor gehören dazu. Schneider-Anwendern, die bisher bei ihren Kommunikationspartnern ein wenig belächelt wurden, können jetzt gut mithalten.
Und Funktionen wie ein eingebautes Notizblatt sowie Modifikationsroutinen zur Wandlung von Binärfiles in sendbare ASCII-Dateien sind Features, die nicht zum alltäglichen Geschäft der Datenfernübertragungsprogramme gehören.
Online sind CPC-Anwender ja allemal (wer will das abstreiten?), aber die Verbindungen zur restlichen Computerwelt schaffen erst solche Programme wie Telecom. Loggen Sie sich ein! (TM)





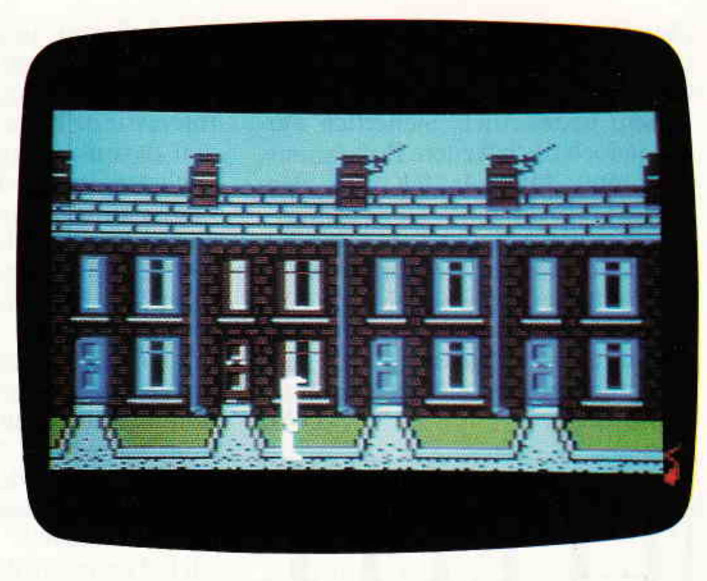
Frankie goes to Hollywood

Hersteller: Ocean
Vertrieb: Fachhandel
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: 29,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

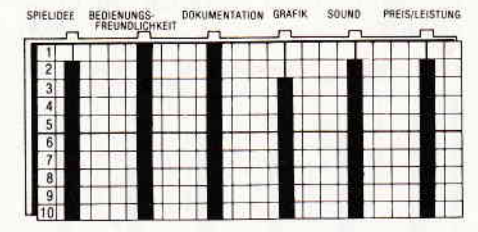
Nachdem die Softwareindustrie, ständig auf der Suche nach neuen Ideen, sich durch die Archive der Film- und

Fernseh-Produzenten gearbeitet hat, so manchem Buch zu einer Wiedergeburt als Computerspiel verhalf, scheint man in deren Ideenabteilungen nun auf die aktuellen Hitparaden aufmerksam geworden zu sein. Denn mit FRANKIE liegt das erste Spiel vor, dessen Handlung geradewegs aus der Philosophie der gleichnamigen Gruppe zu sein scheint. Wer unter uns, der jünger als 25 Lenze, kennt sie nicht, die englische Popgruppe "Frankie goes to Hollywood". Eine Truppe, die mit ihren manchmal wahrlich nicht jugendfreien Songtexten, nicht nur die Gemüter in ihrem Heimatland erhitzen. Von "Relax" bis "Pleasure Dome" reicht auch das Spektrum dessen, was in diesem Spiel abläuft. Nachdem man einsam und verlassen inmitten einer verlassenen, eintönigen Siedlung erwacht ist, beginnt die Suche nach etwas, das schwer zu definieren ist. Zum einen scheint man kein richtiger Mensch zu sein, eher so etwas wie eine Unperson. Ein Mensch, der auf einer völlig anderen Wellenlänge existiert. Zwar ist alles Materielle höchst real, doch völlig leblos. Man muß also versuchen, ein echter Mensch zu werden. Dies geschieht durch das Einsammeln sogenannter "Pleasure Pills". Diese vermindern die Schmerzen, die man als Unperson zu erleiden scheint, und verschaffen dem bedauernswerten Negativmenschen Befriedigung. Je mehr man die Schmerzen verringert und die Befriedigung erhöht, desto realer wird die Person, die außerdem nur noch von einem Gedanken beherrscht wird. Die Suche nach dem sagenhaften "Pleasure Dome", einem Traum, der sich wie ein Magnet an das Bewußtsein der Unperson geheftet hat. Ein Ort, an dem alle Wünsche wahr werden. Zu allem Überfluß stolpert man bei seiner Suche schon bald über eine Leiche, die im Inneren einer der Häuser liegt. Von diesem Moment an bekommt man immer wieder Hinweise, aus denen man sich den Täter herauspuzzlen muß. Doch als sei die Geschichte nicht schon verworren genug, sind da auch noch eine ganze Reihe von eigenartigen Videokassetten, die für die Unperson die Eigenschaft haben, Fenster zu öffnen. Diese sind Durchlässe zu Räumen, die eigentlich an ganz anderer Stelle liegen. Dort warten kniffelige Aufgaben auf den leidgeplagten Unmenschen. Mit nichts, als seinem Körper als Schläger durch ein überdimensionales Break Out, durch Computerräume, Zimmer in denen man beschossen wird, und der Programmierer alleine weiß, was sonst noch auf einen wartet.



Mit FRANKIE GOES TO HOLLYWOOD kommt eines der gelungensten Arcade-Adventures zu uns nach Deutschland. Erfrischend nicht nur der gelungene Spielverlauf, in dem man in die verschiedensten Situationen geraten kann, sondern auch die Idee, die hinter dem Programm steht. Da das Ganze noch mit einer fetzigen Musik unterlegt wurde, wird FRANKIE sicherlich seine Freunde finden.

In einer gewissen Art und Weise ist versucht worden, die Story und das Programm für ein Computerspiel anders als bisher zu entwickeln. Bei den meisten Spielen hat man versucht, eine Geschichte zu erzählen – und sei es nur die simple Flucht eines Bergmannes durch 20 Stockwerke, zurück ans Tageslicht. Nicht so bei Frankie; hier wurde als Handlungsrahmen ein Musikgefühl und die krude Philosophie von Frankies Songtexten bemüht. In dieser Hinsicht liegt hier ein Spiel vor, dessen unbändiger Ideenreichtum



nicht in ein nur schlecht passendes Schema eingesetzt werden mußte. Facettenreich, real und irreal zugleich, präsentiert sich hier erstmals die Subkultur einer Generation, die von tausend Eindrücken lebt, und selbst tausend Eindrücke ist. Begleitet von der fetzigen Musik, macht es dem einen nicht nur Spaß gewissermaßen mit dem Spiel als Spiegel auf sich selbst zu schauen, auch die, die dieses Lebensgefühl nicht selbst kennen, werden von Frankie begeistert sein.
Wirklich ein Spiel für einsame Inseln.
(HS)

Three Weeks in Paradise

Hersteller: Micro Gen
Vertrieb: Fachhandel
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: ca. 35,– DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Wieder einmal gibt es ein Kapitel aus der Familienchronik der "Wally Week Familie". Nachdem man diese schon in "Pyjamarama" und "Everyone is a Wally" bewundern konnte, wird mit "THREE WEEKS IN PARADISE" eine Seite aus dem Ferienalbum dieser Familie aufgeblättert. Eigentlich war ein ganz normaler Südseetrip geplant, aber versehent-



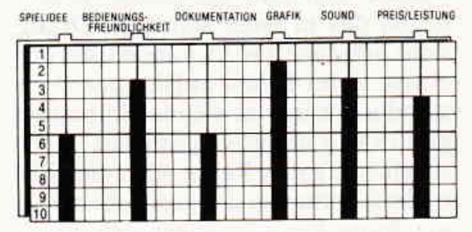
Daß die paradiesische Südseeinsel doch nicht ganz der Garten Eden ist, als der er im ersten Moment erscheint, wird der Week-Familie sehr schnell klar. Eingeborene, deren Leibspeise ausgerechnet Engländer sind, nehmen Wallys Frau Wilma und seinen Sohn Herbert gefangen. Wally gerät dadurch ganz gehörig in Kalamitäten. Kann er seine Lieben retten oder werden die Eingeborenen endlich mal wieder richtig satt? Das ist hier die Frage.

lich wurden sie nicht auf der eigentlichen Ferieninsel abgeliefert. Ohne daß auch nur irgend jemand eine Ahnung davon hätte, befinden sich die Weeks auf einem Eiland, das von wilden Eingeborenen bewohnt wird. Wallys Frau Wilma und sein Sohn Little Herbert werden dann auch prompt von den Eingeborenen entführt und in einen großen Kochtopf gesteckt, auf daß sie schnell gar würden. Da Wally die Seinen natürlich nicht im Stich lassen kann, macht er sich daran, einen Plan für deren Befreiung zu schmieden.

Wie eingangs schon einmal bemerkt, sind die Weeks ja keine Unbekannten mehr, und so ist es irgendwie auch nicht verwunderlich, daß sich THREE WEEKS IN PARADISE fast überhaupt nicht von seinen Vorgängern unterscheidet. Wieder einmal muß man durch das Lösen von Puzzles ein Problem bereinigen. Diese Puzzles sehen folgendermaßen aus:

Überall in den einzelnen Bildern liegen Gegenstände herum, die nacheinander eingesammelt werden müssen, um mit ihrer Hilfe über gewisse Situationen Herr zu werden. Beispielsweise ist es möglich, daß Ihnen auf Ihren Exkursionen auf der Insel ein Krokodil begegnet, das ein weiteres Vordringen verhindert. Möglicherweise ist diesem Tier mit Fleisch oder etwas Ähnlichem so beizukommen, daß es sie dann passieren läßt. So ist eigentlich das gesamte Programm aufgebaut. Die Anforderungen, die hier gestellt werden, sind eher Denkprobleme. Eine präzise Handhabung des Joysticks ist nur in den wenigsten Fällen notwendig. Der einzige Punkt, bei dem sich seit dem ersten Wally-Week-Spiel merklich etwas verändert hat, ist die Grafik, denn die hat sich um einiges verbessert. Sowohl Hintergründe als auch die beweglichen Figuren sind außerordentlich gut gelungen. Im Multicolormodus programmiert und dementsprechend farbenfroh, laufen da Kannibalen und anderes Gesockse umher. Die Hintergründe sind teilweise animiert worden, so daß ein Krokodil, sobald man auf der Bildfläche erscheint, lüstern mit den Kiefern klappert.

Wem die Weeks ans Herz gewachsen sind, dem wird auch dieser neue Streich der englischen Chaotenfamilie gut gefallen. Wer noch nie das Vergnügen hatte, dem wird hier die Gelegenheit geboten, das Versäumte mit dem besten Programm der Serie nachzuholen. Demjenigen, dem die ersten Spiele dieser Art schon nicht gefielen, der kann sich auch die drei Wochen Paradies sparen, weil er auch hier genauso wie mit den anderen Weekiaden bedient wird. (HS)



Crafton & Xunx

Hersteller: ERE Informatique
Autor: Remi Herbulot
Vertrieb: Fachhandel
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: Kass. ca. 39,–; Disk. ca. 59,– DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Ausgesprochen überrascht war ich, daß ein Programm von einer derart hohen Qualität ausgerechnet aus Frankreich zu uns nach Deutschland kommt. Gemeinhin ist Frankreich ja eher bekannt für seine ausgesprochen gute Küche. Daß es dort Menschen gibt, die nicht nur gut kochen können,

beweist dieses Programm. Wenn ich Ihnen an dieser Stelle nun verrate, daß es sich bei Crafton & Xunx um ein Arcade-Adventure mit einem 3D-Szenario, das in einer schrägen Draufsicht präsentiert wird, handelt, werden einige unter Ihnen aufstöhnen. Verständlich wäre das allemal, immerhin gibt es von derartigen Spielen inzwischen eine ganze Reihe. Aber dieses ist anders, denn von der grafischen Aufmachung kann es sogar den Spieljuwelen von Ultimate das Wasser reichen. Wer allerdings nun glaubt, er müsse wie bei Alien 8 oder Nightshade mit nur vier Farben vorliebnehmen, der irrt. Dem Programmierer ist es nämlich gelungen, das Spiel in einem Mode zu schreiben, in dem die volle Farbpalette des CPC zur Verfügung steht. Zwar ist die Auflösung nicht so fein, aber einem geschickten Grafiker ist es gelungen, trotz der hardwaremäßigen Einschränkungen, ein echtes Kleinod zu erschaffen. Es ist wirklich eine ganz andere Sache, ein derartiges Spiel in einer solchen Farbenpracht zu genießen. Allerdings gibt es noch einige andere Unterschiede zu den gemeinhin bekannten 3D-Spielen. Wenn Sie beispielsweise in einem Raum etwas verschieben, etwa einen Tisch, und verlassen danach den Raum und betreten ihn dann nochmals, steht dieser Tisch immer noch an der Stelle, an die Sie ihn vorher gerückt haben. Daraus ergeben sich auch Konsequenzen für den Spieler. Muß man doch genau darauf achten, was man tut. Schiebt man einen Tisch, den man später noch braucht, um an einen höhergelegenen Gegenstand heranzukommen, in eine Ecke des Raumes, bekommt man ihn aus dieser nicht mehr heraus.
Wie eingangs schon erwähnt, handelt es sich bei Crafton & Xunx um ein Arcade Adventure, und in der Natur dieser Spiele liegt es, daß man, um es durchzuspielen, einige Puzzles lösen muß. Diese Denkaufgaben, „was tue ich nun mit diesem Gegenstand", sind gar nicht so einfach. Die Aufgabe, die es zu erfüllen gilt, hat es ebenfalls in sich. Crafton ist nämlich ein Spezialandroide, der in ein schwer bewachtes Computerzentrum eindringen soll. Zusammen mit seinem Sidekick Xunx, einem Lebewesen, das scheinbar nur aus einem Fuß und einem übergroßen Auge besteht, muß er zu den Wissenschaftlern, die in dieser Anlage arbeiten, vordringen. Von ihnen erhält er jeweils einen Teil eines Codes, mit dem



CRAFTON & XUNX
Ein wundervolles dreidimensionales Spielfeld in allen möglichen Farben – das gab es in dieser Qualität bisher nirgends zu sehen. Dieses französische Programm, das in England übrigens unter dem Titel "GET DEXTER" vermarktet wird, wird sicherlich noch einigen Wirbel machen. Bei einigen englischen Software Häusern wird man sich darüber gewiß schon Gedanken machen.

SPIELIDEE | BEDIENUNGS-FREUNDLICHKEIT | DOKUMENTATION | GRAFIK | SOUND | PREIS/LEISTUNG
1 2 3 4 5 6 7 8 9 10

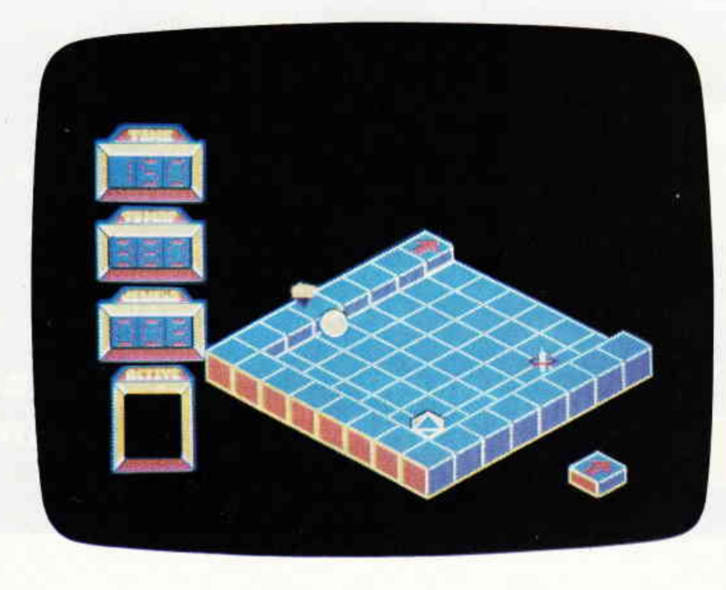
das Computerzentrum lahmgelegt werden kann. Natürlich binden Ihnen die Herren Professoren diese Geheiminformationen nicht freiwillig auf die Nase. Aber sie sind bestechlich, und Crafton muß herausbekommen, mit welchem Gegenstand er die einzelnen Wissenschaftler umgarnen kann. Das ist, soviel sei verraten, keine leichte Aufgabe. (HS)

Spindizzy

Hersteller: Electric Dreams
Vertrieb: Fachhandel
Bedienung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: Kass. ca. 39,–; Disk. ca. 49,– DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Nachdem man Millionen und Abermillionen in alle möglichen Raumfahrtprogramme gesteckt hat, werden eines Tages die Wissenschaftler dieser Welt durch eine sensationelle Nachricht aus ihrer Arbeit geschreckt. Es ist gelungen, eine neue Welt zu erschließen, doch nicht quer durch den Weltraum führt der Weg dorthin, sondern diese Welt befindet sich sozusagen in der Dimension nebenan. Zwar eignet sich die neuentdeckte Welt nicht für Menschen, birgt aber Schätze von unermeßlichen Ausmaßen. Denn dort existieren die für Menschen fast universell einsetzbaren Schwingungskristalle. Der einzige Weg an diese heißbegehrten Juwelen heranzukommen, besteht darin, eine unbemannte Sonde in jene fremde Dimension zu versetzen und mit deren Hilfe die Kristalle einzusammeln. Nachdem es gelungen war, ein solches Instrument zu entwickeln, brauchte man nur noch einen Menschen, der in der Lage ist, die SPINDIZZY-Sonde via Fernsteuerung zu bedienen. Nachdem auch dieser Punkt geklärt war, konnte die Operation beginnen.
Nachdem Spindizzy durch das Dimensionstor in jene andere Zone versetzt worden war, präsentierte sich den Wissenschaftlern an ihrer Fernsteue-

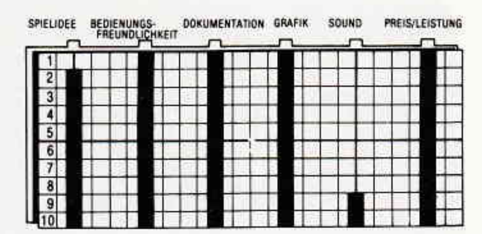


Ein waschechter MARBLE MADNESS Clou ist es schon, dieses SPINDIZZY, aber was für einer. Denn erstens wachsen diese nicht auf Bäumen, und zweitens, braucht sich ein Programm mit solchen Qualitäten nicht zu verstecken. Über 380 verschiedene Bilder, erstklassige Grafik und ein blitzschneller Bildschirmaufbau, das sind Maßstäbe, an denen man nicht vorbeikommt.

rung ein phantastischer Anblick. Eine Welt, die nur aus geraden, ebenen Flächen besteht, breitet sich vor ihnen auf den Monitoren aus. Fraktale Strukturen bilden Brücken und Rinnen, eigenartige Sprungschanzen ermöglichen das Überspringen von Abgründen. Aufzüge, mit denen es möglich ist, zwischen einzelnen Ebenen der fremden Dimension hin und her zufahren, scheinen die Überbleibsel einer längst ausgestorbenen Zivilisation zu sein. Anfänglich ist das Manövrieren der Sonde noch recht einfach, doch schon nach wenigen Minuten dringt Spindizzy in Bereiche der fremden Dimension vor, die den Wissenschaftlern die Haare zu Berge stehen lassen. Immer öfter sieht man sich mit Situationen konfrontiert, die ein Weiterkommen auf den ersten Blick unmöglich erscheinen lassen.

Zweifelsohne wird so mancher Spieler, der sich mit Spindizzy beschäftigt, bemerken, daß es sich hierbei um eine nicht als neu zu bezeichnende Spielidee handelt. Als Urvater dieses Spieles kann man getrost sagen, mußte das allseits gelobte "Marble Madness" herhalten. Abgesehen davon ist Spindizzy auch nicht die erste Umsetzung dieser Spielidee für die CPC's, "Gyroscope" schlug in dieselbe Kerbe und mußte, solange es konkurrenzlos dastand, auch als die definitive Marble Madness-Adaption für die Schneider-Rechner angesehen werden. Aber jetzt ist Spindizzy da. Es zeichnet sich durch eine exquisite und blitzschnelle Grafik aus. Über 380 verschiedene Screens garantieren eine lange Zeit, in der Spindizzy viel Freude und Unterhaltung bereitet. Außerdem ist es so angelegt, daß man anfangs mit recht einfachen Problemen konfrontiert wird; doch je weiter man in die phantastischen, fremden Gefilde eindringt, desto haarsträubender wird es. An eine Funktion, die einem die Orientierung in dieser Dimension erleichtert, wurde ebenfalls gedacht. Mittels Tastendruck läßt sich eine Landkarte aufrufen, in der alle bereits besuchten Orte kenntlich gemacht wurden; die noch nicht besuchten Bilder sind allerdings ebenfalls eingezeichnet. So kann man erkennen, in welche Richtungen man sich bewegen muß, um in noch unbesuchte Regionen zu gelangen.

Spindizzy ist ein rundherum gelungenes Spiel, zu dem man die Jungs von Electric Dreams nur beglückwünschen kann. Als einziges echtes Minus präsentiert sich, wie so oft, auch hier der Sound. Abgesehen von den Arbeitsgeräuschen der Aufzüge, und dem unerfreulichen Geräusch einer zerplatzenden Spindizzy-Sonde ist nichts Interessantes zu hören. (HS)



The Way of the Tiger

Hersteller: Gremlin Graphics
Vertrieb: Fachhandel
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: 100 % M-Code
Preis: 49,– DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Nachdem wir uns als Freunde fernöstlichen Kampfsports in den Weg der explodierenden Faust einführen ließen, als kämpfende Krieger im alten Ägypten einen Namen gemacht haben, und alle Finessen des "Yie Are Kung Fu" aneigneten, kommt nun Gremlin Graphics daher und präsentiert mit "Way of the Tiger" eine kleine Einführung in die Kunst des Ninja Kampfes.

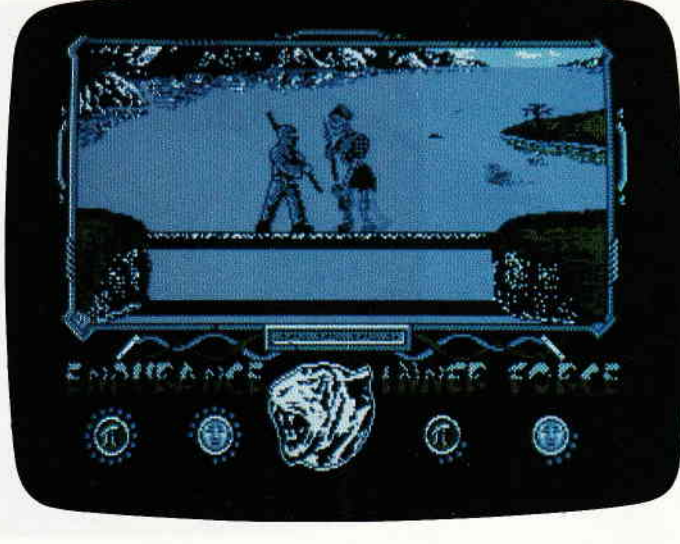
Nun, schlecht war die eingangs erwähnte Kost nun wirklich nicht, also was ist an WOT, das die anderen nicht haben. Nun, alles ist versammelt, um einen zu verdreschen. Zuerst müssen Sie Ihre Qualitäten als Kung Fu-Experte unter Beweis stellen. Allerdings sind die Gegner, die Ihnen gegenüberstehen, wirklich sehenswert. In einer ausgezeichneten Kulisse, die auf drei Ebenen scrollt, stehen Ihnen Ninjas, dolchbewehrte Kobolde, Nashornmenschen, riesige Trolle und "was weiß ich noch" gegenüber. Ist es Ihnen dann gelungen, alles was da kreucht und fleucht zu vertrimmen, haben Sie sich in der Disziplin des unbewaffneten Kampfes soweit qualifiziert, um in die zweite Runde zu kommen.

Hier gilt es zum einen auf einem schmalen, schlüpfrigen Balken zu balancieren, und zum anderen die verschiedensten Gegner mittels eines langen Stabes in das unter dem Baumstamm fließende Flüßchen zu befördern. In diesem Teil des Spieles ist das Auftreten des ersten Widersachers eine wirklich sehenswerte Angelegenheit. Dieser erste Kontrahent, ein halbvermodertes Skelett, das die Reste einer Rüstung trägt, springt, scheinbar aus Einzelteilen bestehend, aus dem Wasser heraus und fügt sich dann auf einem Baumstamm zu einem Ganzen zusammen. Und auch hier gilt der Grundsatz: sind genügend Gegner ins Wasser gefallen, ist man fit genug, in das dritte Level überzugehen. Hier wird der jahrhunderten Kunst des Schwertkampfes gehuldigt. In einer Arena-artigen Umgebung wird man mit wahren Meistern dieser Kunst konfrontiert.

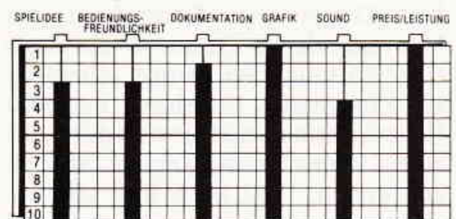
In jeder der drei Disziplinen des Spieles stehen eine Vielzahl von Bewegungen für Verteidigung und Angriff zur Verfügung. Einmal alle möglichen Joystickrichtungen, und dasselbe noch einmal, allerdings mit gedrückter Feuertaste. Insgesamt kann man also 16 verschiedene Bewegungen ausführen; fast eine Selbstverständlichkeit, daß die optimale Beherrschung der Kämpfer erst nach einiger Übung erreicht werden kann. Von der Qualität der Grafik läßt sich sagen, daß "The Way of the Tiger" wohl die beste und originellste von allen Kampfsimulationen hat. Allerdings kommt das auf dem Cover so heiß angekündigte Triple Scrolling, vom ersten Teil mal abgesehen, ein wenig zu kurz.

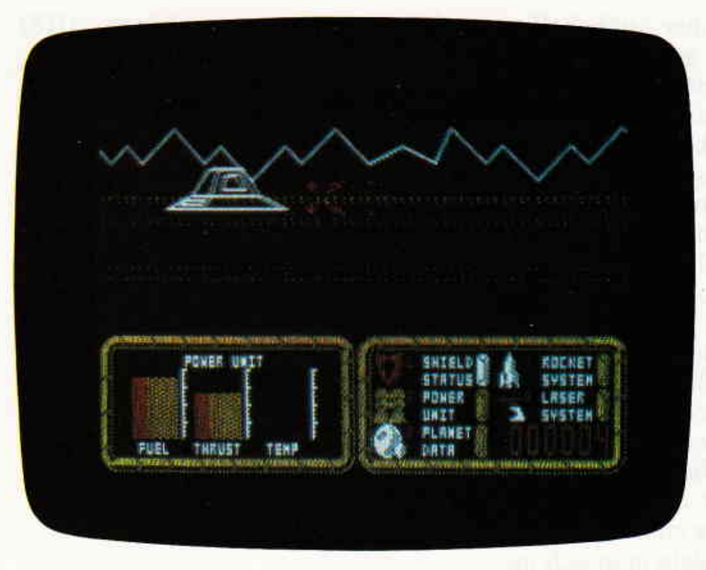
Irgendwie ein wenig unverständlich, lassen sich damit doch offensichtlich wirklich erstklassige Effekte erzielen. Auch vom Sound her ist das Programm

Nebenstehend sehen Sie ein Beispiel für die gelungene Gestaltung und Animation bei WAY OF THE TIGER. Ausgesprochen gelungen ist das Triple-Scrolling, bei dem die Bewegungen quasi auf drei Ebenen stattfinden. Die beiden Kämpfer stellen die erste dieser Scrollebenen dar. Die beiden anderen wurden dazu verwendet, den Hintergründen Leben einzuhauchen.



akzeptabel. Bei jedem Spielanfang kann es mit einem wirklich fernöstlich klingenden Thema aufwarten, und beim Kämpfen kracht es auch ganz anständig.
Komplett betrachtet ist "Way of the Tiger" die umfassendste Simulation von Kampfarten, und man kann getrost behaupten, wenn man dieses Programm hat, kann man sich einige andere sparen. (HS)





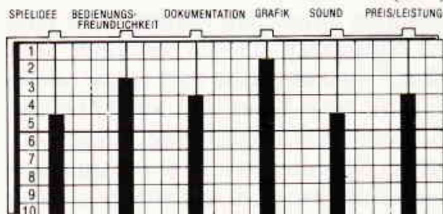
Hier sehen Sie einen Schweber. Mit diesen Invasions-Fahrzeugen dringen Zoltars Truppen auf die Oberfläche der einzelnen Planeten vor. Wenn Sie Ihr eigenes Raumschiff auf einem Planeten reparieren oder auftanken lassen wollen, bleibt es nicht aus, daß Sie solchen Kampfmaschinen begegnen und sich mit ihnen auseinandersetzen müssen.

BATTLE OF THE PLANETS

Hersteller: Micro Gen
Vertrieb: Fachhandel
Steuerung: Joystick
Monitor: Farbe/Grün
Programm: M-Code
Preis: ca. 35,- DM

CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Oh Schreck! Oh Graus! Zoltar bläst den Planeten das Lebenslichtlein aus. Ja, es ist wahr. Zoltar, der Grausliche, und seine fürchterliche Flotte schicken sich an, all den artigen Planeten und Raumstationen das Leben schwer zu machen. Ohne Vorwarnung dringt er in fremdes Hoheitsgebiet ein und feuert auf alles, was da kreucht und fleucht. Ganz klar, daß sich die angegriffenen Planeten zu einer Allianz zusammenschließen, um dem garstigen Zoltar einmal anständig auf seine machthungrigen Griffel zu hauen – bildlich natürlich. Denn im Spiel nachher geht es nicht ums Verhauen, sondern vielmehr darum, wer den anderen als erstes in einzelne Atome zerbläst. In einer superschnellen Vektorgrafik dargestellt, rauschen die Raumschiffe um einen herum. Wer hierbei mehr als einen Blumentopf gewinnen möchte, ist gut damit beraten, viel zu üben. Denn "Battle" ist wirklich gewöhnungsbedürftig. Außerdem reicht es nicht, nur alles im Weltraum Umherfliegende zu zerstören, sondern man muß auch immer darauf achten, wieviele Landungsfahrzeuge der Gegner bisher auf den einzelnen Planeten abgesetzt hat. Wenn man nun schon selbst einige Treffer abbekommen hat, so empfiehlt es sich, einen Planeten anzufliegen, denn dort sind die beiden Basisschiffe, auf denen man seinen intergalaktischen Engelsmacher wieder instandsetzen und volltanken lassen kann. Und wenn man dann schon einmal hier unten ist, kann man auch gleich alle gegnerischen Landefahrzeuge in Asche verwandeln. Hat man erst einmal herausbekommen, wie die Steuerung und das Aufrufen der einzelnen Statusanzeigen funktioniert, läßt sich "Battle" gut und kurzweilig spielen. Allerdings ist die sehr schnelle Vektorgrafik der Weltraumsequenzen echt gewöhnungsbedürftig. Irgendwie wird man den Eindruck nicht los, daß die Programmierer zu lange Elite gespielt haben, oder gar die eine oder andere Routine daraus geklaut haben. Über den allgemeinen sittlichen Nährwert von derartigen Ballerspielen haben wir uns schon lang und breit ausgelassen, auch hier ist dem nichts hinzuzufügen. Aber wenigstens machts Spaß, und das ist doch schon mal was. Wie man allerdings den in der Anleitung erwähnten Rekord von 485339 erreichen oder gar überbieten soll, ist mir ehrlich gesagt ein Rätsel. (HS)



YIE AR KUNG FU

Hersteller: Imagine
Vertrieb: Fachhandel
Steuerung: Joystick/Tastatur
Monitor: Farbe/Grün
Programm: M-Code
Preis: ca. 45,- DM

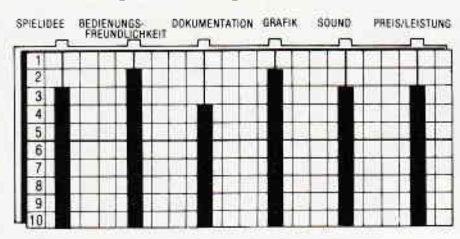
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Super, dachte ich mir, als ich Yie Ar Kung Fu das erste Mal in die Hände bekam. Endlich einmal ein Spiel, das keine ellenlange Vorgeschichte hat. Denn hier geht es einzig und allein darum, Oolong, einem kleinen schlitzäugigen Schlagetot, dabei zu helfen, ein Meister im "Leute verhauen" zu werden. Für Oolong, dessen Vater schon ein Meistervertrimmer war, ein durchaus gerechtfertigtes Anliegen. Aber so leicht ist das nun doch nicht.



Hier prügeln sich gerade Herr Oolong San und Herr Buchu San nach bestem Können. Noch nie war es lustiger sich gegenseitig ein blaues Auge und eine dicke Lippe zu verpassen. Meterhohe Sprünge, gemeine Kicks, und alle nur vorstellbaren fiesen Tricks werden hier benötigt, um mit den einzelnen Gegnern fertig zu werden.

Stehen da doch schon acht andere Meisterschläger bereit, unserem kleinen Oolong anständig eine "reinzunoggern", frei nach dem Motto: „'s gibt nix Schöneres, als wenn der Schmerz nachläßt". So tritt man in den Ring und du schlägst mich, ich schlag dich, schlagen wir uns fürchterlich. Nun ist es ja so, daß es sich hierbei auch nicht gerade um die neueste Prügelsimulation handelt. Aber wird man bei den anderen Spielen, die sich die fernöstlichen Kampfkünste zum Thema gemacht haben, eher an die Übertragung der Karate Weltmeisterschaft vor zwei Jahren erinnert, so kommen einem hierbei Filmtitel wie "Die Todesfaust des Stink Fing" oder "Die 13 Ködel des Gelben Drachen" in den Sinn. Denn Unfairness und Hinterlist werden hier groß geschrieben. So ist zum Beispiel der einzige Gegner, der sich, außer seiner Extremitäten, keiner anderen Waffe bedient, der erste Widersacher. Schon der zweite Gegner, der Packungsgrafik nach ein zuckersüßes Mädel, macht einem das Leben durch gemeine Wurfsterne schwer. So geht es quer durch das fernöstliche Waffenarsenal, und alles, was sich zwischen Stock und Nunchacku dazu eignet, dem Gegner blaue Flecken zu verpassen, wird hervorgekramt und angewandt. Dem Spieler stehen zur Verteidigung zehn Tritte, Schläge und Kicks zur Verfügung, einer fieser als der andere. Angenehm ist diese Vielzahl von möglichen Verteidigungsaktionen allemal, erleichtern sie das Überleben doch ungemein. Dadurch, daß Yie Ar Kung Fu die Sache eher Comic-haft präsentiert, was alleine schon durch die mehrere Meter hohen Sprünge der Kämpfer deutlich wird, kann man bei dieser rohen Prügelsimulation riesig viel Spaß haben. Besonders, wenn man zu zweit spielt. Sollte Ihr Vater Ihnen einmal eine drastische Erziehungsmaßnahme auferlegt haben, die in keinem Falle Ihre Zustimmung findet, laden Sie Ihn doch zu einer Partie Kung Fu ein. Hier können Sie dann, ausreichendes Training vorausgesetzt, dem Herrn Erziehungsberechtigten mal anständig Mores lehren. Ist gut für den Frust und entspannt ungemein. (HS)



DESERT RATS

Hersteller: CCS
Vertrieb: Fachhandel
Steuerung: Joystick+Tastatur/Tastatur
Monitor: Farbe/Grün
Programm: Basic/M-Code
Preis: ca. 39,– DM

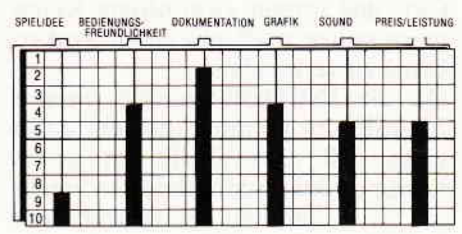
CPC 464 ☒ CPC 664 ☒ CPC 6128 ☒

Mehr als vierzig Jahre liegen die Greuel und Schrecken des Zweiten Weltkrieges nun zurück. Eine Auseinandersetzung, deren Resultate Menschen in aller Welt bis zum heutigen Tage heimsuchen. Millionen Tonnen von Blut und Tränen überschwemmten in diesen Tagen Europa; ein Strom von Leid, der bis zum heutigen Tage nicht versiegt ist. Kaum ein Krieg hatte eine nachhaltigere Wirkung auf die Menschheit, sei es aus wirtschaftlicher oder aus politischer Sicht. Und obwohl seither nun doch so viele Jahre vergangen sind, darf keiner den Fehler begehen und diese Zeit sowie die politischen Umstände, die zum Zweiten Weltkrieg geführt haben, zu vergessen. Ungeachtet allerdings der sechs Millionen Menschen, die damals ihr Leben verloren, betreibt ein englisches Softwarehaus Vergangenheitsbewältigung in einer sehr sonderbaren Form. Richtig ist zwar, daß der Afrikafeldzug in diesem globalen Konflikt bei weitem einer der am "humansten" geführten Feldzüge war, trotzdem empfinde ich es als Hohn, über tausende Gefallene ein solches Ereignis zum Thema einer abstrakten Strategie-Simulation zu machen. Wahlweise kann man hierbei aus verschiedenen Komponenten – in bester Feldherrenmanier – einen kleinen Krieg zusammenbasteln.

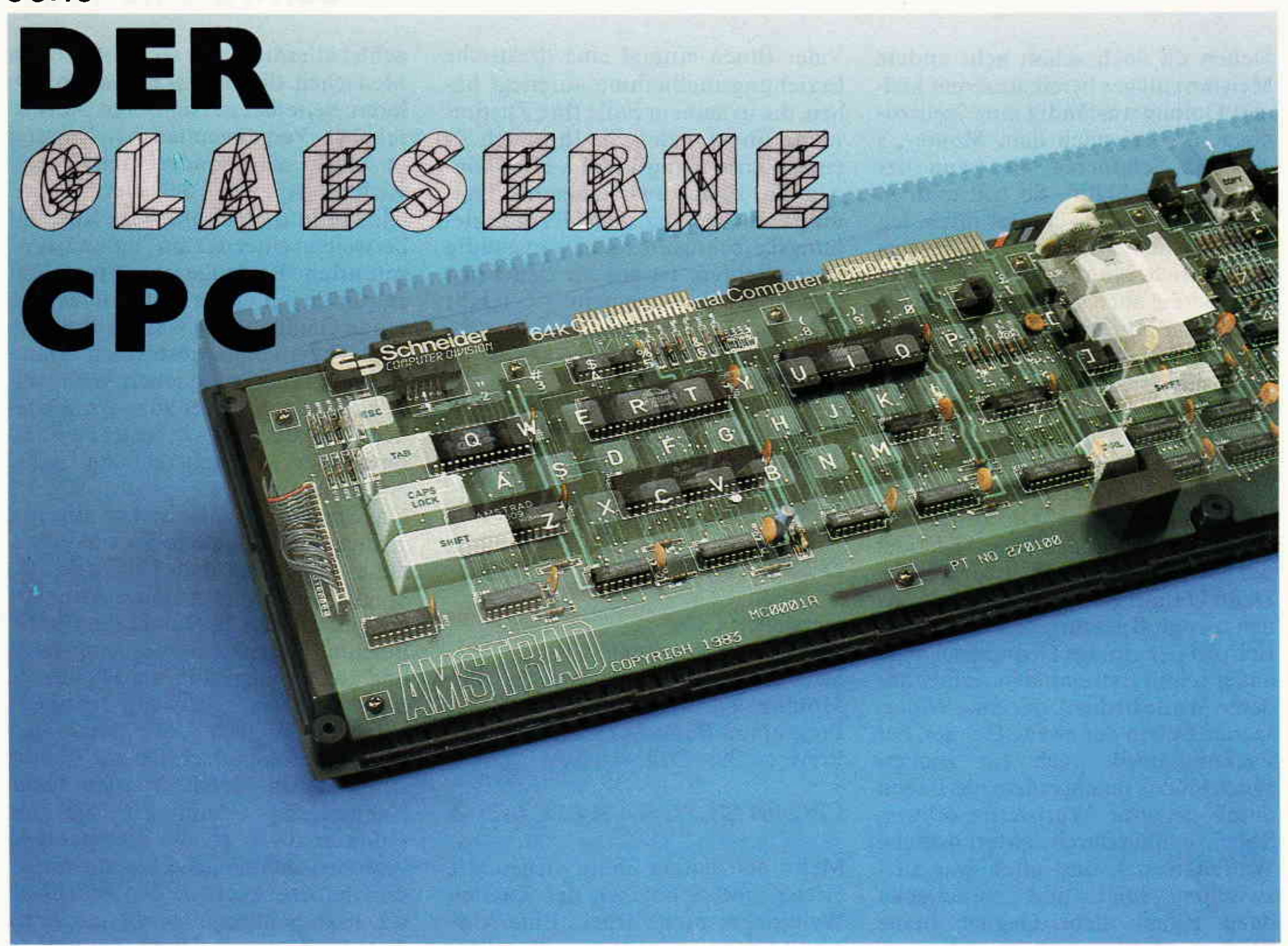
Das Spiel selbst ist sicherlich eine der gelungensten Strategie-Simulationen, die es bisher auf dem CPC gab. Als Spieler verteilt man seine Truppen und gibt ihnen im Laufe des Spiels Befehle. Marschbefehle, eingraben und lauern, angreifen, zurückziehen, sogar das Verschiffen von Truppenteilen ist möglich, sofern man geeignete Transportschiffe hat. Realistisch bis in Details werden sogar Nachschübe erteilt, die Sie als Feldherr dann an die Truppenteile vergeben müssen, die Sie am nötigsten haben. Auch in der Anleitung war man bemüht, die größtmögliche Autentizität zu erreichen. Zusätzlich, zu einer sehr ausführlichen Beschreibung des Spiels, ist die Hälfte des Heftes angefüllt mit historischen Daten über eben jenen Feldzug. Kneife ich als Reviewer nun beide Augen zu, bleibt ein Spiel übrig, das all denen, die sich gerne mit kniffeliger Strategie auseinandersetzen, in "Desert Rats" eine wirklich gut gelungene Simulation vorfinden. Spieltechnisch wie grafisch ist es dem Programmierer hier gelungen, ein überdurchschnittlich gutes Spiel abzuliefern. Öffne ich allerdings dann wieder beide Augen, bleibt mir nichts anderes, als zu feststellen, daß es doch geschmacklos ist, ja fast schon gefährlich anmutet, solche von Trauer und Elend geprägten Abschnitte unserer Geschichte in ein Computerspiel zu packen. Der junge Mann, der dieses Spiel geschrieben hat, wäre sicherlich besser damit beraten gewesen, sich ein abstrakteres Thema für seine Konfliktsimulationen zu suchen.

(HS)

Zwar ist es richtig, daß die Kampfhandlungen in DESERT RATS bis zum "Geht-nicht-mehr" abstrahiert wurden. Trotzdem hat man sich ein wirklich makaberes Thema herausgesucht. Auf die Berge von Toten, die der Zweite Weltkrieg hinterlassen hat obenauf noch ein Computerspiel zu setzen, kann man getrost als Geschmacklosigkeit bezeichnen.



DER GLAESERNE CPC



Auch diesmal beginnen wir mit einem sehr einfachen Projekt: Der Reparatur der DEC$-Funktion.
Wie Sie wahrscheinlich schon wissen, ist sie im CPC 464 zwar vorgesehen, funktioniert aber, aufgrund eines kleinen Programmfehlers, nicht richtig.

Gedacht war es eigentlich so:
DEC$ (< numerischer Ausdruck>, < Formatierungsstring >)
und sollte den numerischen Ausdruck als formatierte Zeichenkette zurückgeben, wobei der Formatierungsstring genau wie bei PRINT USING angegeben wird.
Wenn der CPC nun bei der Berechnung eines Ausdrucks auf eine Basic-Funktion (Code &FF) trifft, so wird routinemäßig überprüft, ob das folgende Argument mit einer offenen Klammer beginnt. Aus unerfindlichen Gründen wird dieser Test zu Beginn der DEC$-Routine aber noch einmal ausgeführt, so daß man beim CPC 464 immer zwei offene Klammern setzen muß, um einen Syntax-Error zu vermeiden.
Da die INCODE-Routine unseres XBASIC-Programms jedoch dafür sorgt, daß die Tabelle der neuen Kommandos und Funktionen bereits vor der internen Standardtabelle durchsucht wird, stellt es kein Problem dar, die DEC$-Funktion mit einem anderen Token zu versehen und neu zu definieren. Zunächst ist wieder eine Eintragung in der BEFTAB-Tabelle fällig:

```
DM "DEC"
DB &A4    ;"$"+&80
DB &20    ;neues Token
```

Hier können Sie die vom CPC 464 nicht benutzten Codes &20 - &39 als Token einsetzen; es können also insgesamt 32 zusätzliche oder geänderte Basic-Funktionen eingebunden werden. Trifft der Interpreter auf einen dieser Codes, so wird die XBASIC-Routine FNBER (Funktionsberechnung) aufgerufen, die den Error abfängt und den CPC dazu bringt, die Adresstabelle FNADR für neue Funktionen zu verwenden. Ähnlich wie bei den Kommandoadressen ist hier die erste Eintragung für das Token &20 zuständig, die zweite Eintragung für den Code &21 usw.
Wir brauchen hier jetzt also nur die Adresse der normalen DEC$-Routine einzutragen, und zwar so, daß der unnötige Test auf "(" einfach übersprungen wird. Im ROM-Listing findet man zu Beginn der Adresse &F8EA die Sequenz:

```
CALL &DD37
DB   &28
```

und da haben wir ihn schon, den Übeltäter! Die Routine &DD37 testet, ob das nächste Zeichen im Basic-Programm mit dem auf den CALL folgenden Data-Byte identisch ist; &28 ist der ASCII-Code für "(". Ist alles in Ordnung, so wird der Ablauf nach dem Prüfbyte fortgesetzt, andernfalls gibt es einen Syntax-Error. Also lassen wir unsere neue DEC$-Funktion einfach 4 Bytes später einspringen, und damit sieht die Eintragung in der Adresstabelle so aus:

```
FNADR DW &F8EE
```

Assemblieren Sie das Ganze einmal probeweise, und Sie werden sehen: jetzt funktioniert DEC$ genauso, wie es sein sollte!
Und da wir gerade so gut in Form sind, können wir gleich eine weitere Basic-Funktion verbessern. Nichts gegen das Schneider-Basic – aber schauen Sie sich doch einmal an, was man auf dem ZX Spectrum mit der VAL-Funktion veranstalten kann:

```
10 LET x=PI/4
20 LET a$="sin(x)"
30 PRINT VAL(a$)
```

Ohne zu zögern, wird der Computer die Formel berechnen und das Ergebnis 0.707107 anzeigen. Damit ist es zum Beispiel möglich, auf elegante Art Funktionen per INPUT in ein Mathematikprogramm einzugeben. Läßt

man dieses Programm jedoch auf dem CPC laufen, so erscheint eine dumpfe Null auf dem Bildschirm; unser gutes Stück kann nämlich mit VAL leider nur Strings bearbeiten, die aus Ziffern bestehen. Doch das soll sich ändern!
Zunächst wieder die übliche Zeremonie: Tragen Sie in die Befehlstabelle:
DM "VA"
DB &CC ;"L"+&80
DB &21 ;neues VAL-Token
ein, und in die Adresstabelle für Funktionen muß dann nach der Adresse für DEC$:
DW VAL ;Adr. der VAL-Routine
folgen. Und jetzt bitte etwas Konzentration – die neue VAL-Routine hat es nämlich in sich: Wir holen uns die zu dem Funktionsargument gehörenden Stringparameter (Adresse, Länge), wandeln den String in den Interpretercode um und lassen dann die Routine "Ausdruck berechnen" darauf los. Abschließend wird noch getestet, ob wirklich etwas Numerisches dabei herausgekommen ist. Soweit der Rohbau – und hier noch ein paar spezielle Punkte, die zu beachten sind:
- Der Interpreter hat bereits geprüft, ob nach VAL eine offene Klammer folgt. Wir müssen sicher stellen, daß am Ende des Ausdrucks eine Klammer zu steht.
- Falls der Ausdruck einen Leerstring ("") ergibt, soll der Wert 0 zurückgegeben werden.
- Für die Umwandlung in den Interpretercode muß der String durch eine 0 als Endmarkierung ergänzt werden. Damit wir aber nicht den Beginn des folgenden Strings damit kaputt schreiben, wird der Inhalt dieser Speicherstelle aufbewahrt und nach der Aktion wieder eingesetzt.

Das Listing "VAL-Routine" demonstriert, wie diese Ideen in die Praxis umgesetzt werden. Falls noch Unklarheiten auftreten sollten, so wird Ihnen die Übersicht "wichtige Interpreter-Routinen" weiterhelfen.
So – und zum Abschluß soll noch ein interessantes Projekt vorgestellt werden, das wir Ihnen dann zur weiteren Bearbeitung übergeben möchten. Es handelt sich um eine komfortable Eingaberoutine für den CPC 464. Zwar stellt das Schneider-Basic bereits den INPUT-Befehl zur Verfügung, doch insbesondere der professionelle Programmierer ist damit nicht recht glücklich. Nur zu leicht passiert es, daß eine zu lange Eingabe oder die Fehlermeldung "Redo from Start", Teile einer sorgfältig geplanten Eingabemaske überschreibt, und außerdem läßt sich der Cursor nicht auf ein bestimmtes Eingabefeld beschränken.
Wie kann man hier Abhilfe schaffen? Die grundlegende Idee ist, eine Basic-Funktion zu definieren, die eine Art Mischung aus INPUT und INKEY$ darstellt und zum Beispiel so aussehen könnte:

a$=GETKEY$ (z)

wobei z die maximale Länge der Eingabe bestimmt. Natürlich soll dabei auch der Cursor und die eingegebenen Zeichen auf dem Bildschirm erscheinen. Nach ENTER wird die Eingabe in a$ übernommen und der Programmablauf fortgesetzt.
Schreiten wir also sofort zur Tat. Die Eintragungen in die Befehls- und Adresstabellen brauchen wir an dieser Stelle wohl nicht mehr zu erläutern – beachten Sie bitte nur, daß es sich hier um eine Basic-**Funktion** handelt.
Die GETKEY$-Routine baut sich wie folgt auf: Zunächst brauchen wir die maximale Länge als 8-Bit-Wert, dann folgt der obligatorische Test auf ")". Danach werden in einer kleinen Schleife, mit Hilfe der Betriebssystem-Routine KM READ CHAR, solange Zeichen aus dem Tastaturpuffer gelesen, bis er leer ist. Diese Maßnahme verhindert, daß bereits vorher eingegebene Zeichen ungewollt übernommen werden.
Die nächsten Schritte sind leicht zu durchschauen: Der Cursor wird eingeschaltet, und die Routine sammelt Zeichen von der Tastatur ein, bis jemand ENTER (ASCII-Code 13) drückt. Falls die maximale Länge erreicht ist, werden keine weiteren Eingaben akzeptiert. Da noch nicht klar ist, wie lang die Eingabe insgesamt sein wird, werden alle Zeichen zunächst in den CPC-Eingabepuffer übernommen.
Interessant wird es in dem Programmteil, der mit dem Label "READY" gekennzeichnet ist. Er demonstriert, wie dem Interpreter ein neuer String zur weiteren Bearbeitung übergeben wird.
Zunächst muß er angemeldet werden, indem wir die Länge der Routine &FC19 übergeben. Sie reserviert den nötigen Platz im RAM und gibt uns die Adresse dieses Bereichs im DE-Register zurück. Danach werden die Zeichen dort eingetragen, und die Routine &FBBA sorgt schließlich für die endgültige Übernahme in die interne Stringverwaltung, die übrigens ziemlich kompliziert ist – leider müssen wir uns hier auf die notwendigsten Hinweise beschränken.

```
;************************ VAL-Routine

VAL     CALL &CEFB   ;Ausdruck holen
        CALL &DD37   ;Test auf
        DB   &29     ;Klammer zu
        PUSH HL      ;Programmzeiger retten
        CALL GETVAL  ;Value berechnen
        POP  HL
        RET

GETVAL  CALL &FBDA   ;Stringparameter holen
        JP   Z,&FF0A;Akkuinhalt übernehmen
        EX   DE,HL   ;HL=Stringadresse
        PUSH HL      ;merken
        LD   D,0
        LD   E,A     ;Länge nach DE
        ADD  HL,DE
        LD   E,(HL)  ;Ende sichern
        LD   (HL),D  ;0 einsetzen
        EX   (SP),HL;HL=Stringadresse
        PUSH DE
        CALL &DEBB   ;-> Interpretercode
        CALL &CEFB   ;Ausdruck berechnen
        POP  DE
        POP  HL
        LD   (HL),E  ;Restore Ende
        JP   &FF2D   ;Test auf numerisches
                      Ergebnis
```

Wenn Sie die GETKEY$-Routine auf die übliche Weise in das XBASIC-System einbinden, assemblieren und initialisieren, können Sie sich mit dem folgenden kurzen Programm von der Funktionsweise überzeugen:

10 CLS:LOCATE 12,12
20 a$=GETKEY$ (5)
30 PRINT a$

Störend macht sich nur noch bemerkbar, daß keine Möglichkeit zur Eingabekorrektur besteht. Hier ein paar Hinweise dazu, wie sich ein Rückwärtslöschen mit DEL einbauen läßt:

- Die DEL-Taste ist mit dem ASCII-Code 127 belegt.
- Das Steuerzeichen CHR$ (8) setzt den Cursor eine Stelle zurück.
- Das Steuerzeichen CHR$ (16) löscht das Zeichen auf Cursorposition.

... und den Rest möchten wir Ihrer Kreativität überlassen, denn damit sind wir am Ende dieser Folge des "Gläsernen CPC" angelangt. Womöglich werden Sie noch den einen oder anderen Hinweis vermissen; aber es ist leicht einzusehen, daß wir nicht alle Feinheiten eines 16 kByte umfassenden Maschinenprogramms im Rahmen dieses Artikels klären können. Er soll hauptsächlich eine Starthilfe geben - und wenn Sie für eigene Projekte noch weitere Informationen brauchen, so nehmen Sie sich am besten ein kommentiertes ROM-Listing vor und suchen sich einen möglichst ähnlichen Befehl als Beispiel; das ist meistens sehr aufschlußreich.

Und jetzt gibt es noch Zweierlei zu vermelden - eine gute und eine schlechte Nachricht. Welche wollen Sie zuerst lesen? Okay, also hier die Schlechte, dann haben wir es hinter uns: Diese Folge ist unwiderruflich die letzte Folge des "Gläsernen CPC", da eine neue Serie in Planung ist, die noch etwas Vorbereitungszeit braucht - aber Sie dürfen jetzt schon gespannt sein!

Allerdings möchten wir uns von Ihnen nicht ohne ein gehöriges Feuerwerk verabschieden, und das ist die gute Nachricht: In diesem Heft finden Sie eine hochkarätige Basic-Erweiterung für den CPC 464, in der alle "Highlights" der vergangenen Folgen des "Gläsernen CPC" zusam-

```
;************************* GETKEY$-Routine

key$    call &ce67      ;8 Bit-Wert holen
        ld   b,a        ;nach B
        call &dd37      ;Test auf
        db   &29        ;")"
        push hl         ;Programmzeiger retten
        call &bb81      ;TXT CURSOR ON
read    call &bb09      ;KM READ CHAR
        jr   c,read     ;Tastaturpuffer leeren
        ld   c,0        ;Zeichenzähler auf 0
        ld   hl,&aca4   ;Start Eingabepuffer
        push hl         ;merken
wait    call &bb06      ;KM WAIT CHAR
        cp   13         ;ENTER gedrückt ?
        jr   z,ready    ;ja, fertig
        ld   e,a        ;Zeichen nach E
        ld   a,b        ;maximale Länge
        cp   c          ;erreicht ?
        jr   z,wait     ;ja, nicht übernehmen
        ld   a,e        ;Zeichen auf Bildschirm
        call &bb5a      ;TXT OUTPUT
        ld   (hl),a     ;und in den Eingabepuffer
        inc  hl         ;Zeiger und
        inc  c          ;Zähler erhöhen
        jr   wait       ;auf Zeichen warten

ready   call &bb84      ;TXT CURSOR OFF
        ld   a,c        ;Stringlänge nach A
        call &fc19      ;Platz reservieren
        pop  hl         ;Start Eingabepuffer
        ld   b,0
        or   a          ;Leerstring ?
        jr   z,leerst
        ldir            ;String übertragen
leerst  call &fbba      ;String übernehmen
        pop  hl         ;Programmzeiger zurück
        ret
```

mengefaßt wurden, nebst einigen zusätzlichen Bonbons, die es gewaltig in sich haben - schauen Sie sich das Programm einmal an!
Wenn Sie Lust haben, diese Erweiterung noch um ein paar eigene Ideen zu bereichern - warum nicht? Die nötigen Grundlagen dazu wurden in dieser Folge besprochen, und es bleibt nur noch die Frage, wie Sie an den Assembler-Quellcode für das Programm kommen. Auf einen Abdruck in diesem Heft haben wir verzichtet, da er enorm viel Platz beansprucht. Und abgesehen davon - das Abtippen der circa 26 kByte wäre sicher auch kein reines Vergnügen.
Doch wozu gibt es schließlich die DATABOX? In der Kassetten- und Diskettenausgabe zu diesem Heft finden Sie den kommentierten Quellcode in Form von REM-Zeilen. Der CPC International-Assembler (Heft 1/86) kann ihn in dieser Form direkt verarbeiten, ansonsten hilft ein Abspeichern im ASCII-Format und notfalls ein kleines Konvertierungsprogramm, um ihn für andere Assembler aufzubereiten.
Und das wär's dann - wir hoffen, daß Ihnen der "Gläserne CPC" viele Anregungen für eine kreative Arbeit mit Ihrem Schneider-Computer gegeben hat!

Wichtige Interpreterroutinen

1. &DEBB String in Interpretercode wandeln
Einsprung: Stringadresse in HL
Aussprung: Startadresse Puffer (&0040) in HL
Das Stringende muß durch ein Nullbyte markiert sein. DE bleibt unverändert.

2. &CEFB Ausdruck berechnen
Beim Einsprung muß in HL der Basic-Programmzeiger stehen.
Beim Aussprung sind die Speicherstellen ab Adresse &B0C1 in Abhängigkeit vom Ergebnistyp wie folgt belegt:
Integer: (&B0C1)=2, ab &B0C2 steht der 2-Byte Integerwert.
String: (&B0C1)=3, ab &B0C2 steht die 2-Byte Adresse des Stringdescriptors.
Fließkomma: (&B0C1)=5, ab &B0C2 steht der 5-Byte Fließkommawert.
Das BC-Registerpaar bleibt unverändert.
Ein ab &B0C1 abgelegtes Ergebnis kann mit folgenen Routinen weiterverarbeitet werden:

3. &FBDA Stringparameter holen
Einsprung: Ergebnis ab &B0C1
Aussprung: Länge des Strings in A,B.
Adresse des Strings in DE
Falls das Ergebnis kein String war, erfolgt ein Fehleraussprung mit "Type Mismatch". Bei einem Leerstring wird das Zero-Flag gesetzt. HL bleibt unverändert.

4. FF2D Test auf numerisches Ergebnis
Einsprung: Ergebnis ab &B0C1
Aussprung: Integer → Carry=1, Integerwert in HL
Fließkomma → Carry=0, Adresse der Fließkommazahl in HL
String → Fehleraussprung mit "Type Mismatch".
In den folgenden Routinen ist die Ausdrucksberechnung bereits enthalten. Falls das Ergebnis nicht numerisch ist oder nicht im angeforderten Bereich liegt, wird "Type Mismatch" bzw. "Improper Argument" ausgegeben. Der Programmzeiger muß sich beim Einsprung in HL befinden und wird beim Lesen des Ausdrucks entsprechend weitergesetzt.

5. &CE91 16-Bit-Wert holen
Aussprung: 16-Bit-Wert in DE
BC bleibt unverändert

6. &CE86 Integerwert mit Vorzeichen holen
Aussprung: Integerwert in DE
BC bleibt unverändert

7. &CE67 8-Bit-Wert holen
Aussprung: 8-Bit-Wert in A
BC bleibt unverändert

8. &C1FB 8-Bit-Wert < A holen
Einsprung: Maximalwert+1 in A
Aussprung: 8-Bit-Wert in A
BC,DE bleiben unverändert
Das Ergebnis einer Basic-Funktion muß dem Interpreter für weitere Berechnungen zur Verfügung gestellt werden, indem es ab Adresse &B0C1 abgelegt wird (siehe 2.). Dabei helfen folgende Routinen:

9. &FF0A Akkuinhalt als Integerzahl übernehmen
Einsprung: Zahl in A
Aussprung: Integerzahl in HL
BC,DE bleiben unverändert

10. &FF0D Integerzahl in HL übernehmen
Einsprung: Integerwert in HL
F,BC,DE,HL bleiben unverändert.
Falls eine Basic-Funktion einen Text zurückgibt, muß er zunächst im Speicher abgelegt werden:

11. &FC19 Platz für String reservieren
Einsprung: Länge des Strings in A
Aussprung: Adresse des reservierten Bereichs in DE.
AF,BC,HL bleiben unverändert.
Nachdem der Text dort eingetragen wurde, wird er mit folgender Routine in die interne Verwaltung übernommen:

12. &FBBA String übernehmen
Einsprung: Keine Bedingung
BC bleibt unverändert
Fehlermeldungen können über die folgende Einsprungadresse erzeugt werden:

13. &CA94 Error ausgeben
Einsprung: Fehlernummer in E
(Siehe CPC-Bedienungshandbuch, Anhang VIII)
Die Folge:
LD E,2
JP &CA94
gibt zum Beispiel einen Syntax-Error aus. Dabei spielt es keine Rolle, in welchem Zustand sich der Z 80-Stack zum Zeitpunkt des Fehleraussprungs befindet, da er ohnehin neu initialisiert wird. (M. Uphoff)

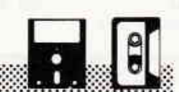
Programmhit des Monats

XBASIC

Erweiterung für den CPC 464

... und das klingt ein bißchen wie „die x-te Basicerweiterung für den CPC 464". Doch lassen Sie sich davon nicht täuschen - hinter diesem unscheinbaren Titel verbirgt sich ein Programm, das es in sich hat! Da es erst knapp vor Redaktionsschluß fertig wurde, präsentieren wir es hier kurzerhand unter seinem Arbeitstitel ("eX-tended BASIC") und verzichten auf eine großartige Verpackung.

für 464



Die neuen Befehle wurden nicht als RSX-Kommandos programmiert, sondern als "echte" Erweiterungen direkt in den Basicinterpreter des CPC eingebunden. Damit entfällt der übliche Strich vor dem Befehlswort, und auch die umständliche Variablenübergabe per "Klammeraffe" ist nicht mehr notwendig. Weiterhin enthält XBASIC erstmalig neue Basic-Funktionen - auch das ist mit der RSX-Methode nicht realisierbar.
Insgesamt bietet XBASIC 20 zusätzliche Befehle, die eine ausgewogene Mischung aus nützlichen Arbeitshilfen und effektvollen Mitteln zur Bildschirmgestaltung darstellen. Hier ein paar Stichworte:

- Pull-Down-Windows
- maximal 3 (!) Bildschirmseiten
- eine sehr schnelle FILL-Routine
- variable Schriftgröße bei Textausgaben
- direktes Editieren von Stringvariablen
- selektives Listen von Programmzeilen
- HEX-Monitor
- zusätzliche Farben
- eingebaute Digitaluhr

Und das ist noch längst nicht alles - die weiteren Möglichkeiten können Sie der Benutzeranleitung entnehmen.
Um das Programm in Ihre Softwaresammlung einzureihen, gehen Sie am besten wie folgt vor:

1. Tippen Sie den Basiclader ab, und sichern Sie ihn vorsichtshalber vor dem ersten Start auf Kassette oder Diskette.
2. Für jeden Programmblock wird separat eine Prüfsumme ermittelt. Ist sie korrekt, so folgt die Meldung "Block x OK"; anderenfalls wird der fehlerhafte Block angegeben. Wenn alle 16 Blöcke in Ordnung sind, wird der Maschinencode automatisch als Binärfile abgesaved (Datenträger bereitstellen!) und danach initialisiert, so daß Sie die neuen Befehle gleich ausprobieren können. Ein zweiter Start des Laders nach der Initialisierung kann Probleme ergeben, verzichten Sie also besser darauf.
3. Schreiben Sie ein kurzes Ladeprogramm für den Maschinencode;

```
10 SYMBOL AFTER 256
20 MEMORY &9DFF
30 SYMBOL AFTER 240
40 LOAD "XBASIC.BIN"
50 CALL &9E00
60 PRINT "XBASIC initialisiert"
```

Speichern Sie es ab, und danach können Sie den Basiclader vergessen.
Hier noch ein Hinweis für Bastler und Assemblerspezialisten: In der Serie „Der gläserne CPC" erfahren Sie, wie die XBASIC-Erweiterung noch ausgebaut werden kann; und noch ein Hinweis für Basicspezialisten: Interessante Programme, die unter XBASIC geschrieben wurden, werden wir natürlich gerne veröffentlichen!

Beschreibung der neuen Befehle:

Die Befehls- und Funktionsargumente, die nicht in Fettdruck erscheinen, sind wahlweise.

CHAR < **ganzzahliger Ausdruck** >
KOMMANDO:
Bestimmt die Größe und den Typ der Zeichen bei Textausgaben. Hier können folgende Werte eingesetzt werden:
0 - normale Schrift
1 - Zeichenhervorhebung ("emphasized")
2 - Dünnschrift ("light")
4 - doppelte Breite ("Mode Mix")
8 - doppelte Höhe
16 - unterstreichen
Die verschiedenen Möglichkeiten können kombiniert werden, indem man einfach die entsprechenden Werte addiert. So ergibt zum Beispiel CHAR 22 unterstrichene Zeichen in der Schriftart "light" mit doppelter Breite (22=16+4+2).
Dieses Kommando wirkt auf alle Textausgaben mit PRINT, WRITE, LIST, INPUT usw., nicht aber auf direkte Tastatureingaben und Ausgaben auf Grafikcursor-Position.

```
10 CHAR 12:PRINT" Gigantisch !"
```

CIRCLE < **Radius** >, < Farbstift >
KOMMANDO:
Zeichnet einen Kreis mit dem angegebenen Radius und Farbstift. Der Mittelpunkt wird durch den Grafikcursor bestimmt.

```
10 MOVE 320,200
20 FOR r=0 TO 100 STEP 10
30 CIRCLE r,3
40 NEXT r
```

COPY$(< # Window >, < **x-Koord.** >,< **y-Koord.** >, < Länge >)
FUNKTION:
Liest eine Zeichenkette vom Bildschirm. Die Textkoordinaten geben die Position des ersten Zeichens an. Wenn keine Länge angegeben wird, so wird nur ein Zeichen gelesen. Nicht identifizierbare Zeichen (z.B. Grafik) werden durch ein Leerzeichen ersetzt.

```
10 LOCATE 10,10:PRINT "Basic"
20 a$=COPY$(10,10,5)
30 PRINT a$
```

CURSOR < **ganzzahliger Ausdruck** >
KOMMANDO:
Ermöglicht es, während eines Programmablaufes den Cursor auf dem Bildschirm erscheinen zu lassen:

CURSOR 1 – schaltet den Cursor ein
CURSOR 0 – schaltet ihn aus
Im Direktmodus hat dieser Befehl keine Auswirkung.

DEC$ (<numerischer Ausdruck>,<Formatierungsstring>)
FUNKTION:
Gibt den numerischen Ausdruck als formatierte Zeichenkette zurück. Die Formatierung erfolgt genau wie bei PRINT USING.

```
10 PRINT DEC$(1/7," #.###")
```

EDT <Stringvariable>
KOMMANDO:
Editiert den in der Stringvariablen enthaltenen Text auf dem Bildschirm, wo er auf die beim CPC übliche Weise bearbeitet werden kann. Nach ENTER wird der veränderte Text wieder in die Stringvariable übernommen.

```
10 a$="Dieser Text soll editiert werden"
20 EDT a$:PRINT a$
```

FILL <Farbstift>
KOMMANDO:
Füllt eine beliebig umrandete Fläche auf dem Bildschirm mit der angegebenen Farbe. Der Vorgang beginnt an der aktuellen Grafikcursor-Position und wird durch das Grafikfenster begrenzt.

```
10 MOVE 320,200
20 CIRCLE 100,2:FILL 3
```

FIND <Textausdruck>, <Zeilenbereich>, <#Ausgabegerät>
KOMMANDO:
Listet alle Zeilen eines Basicprogramms, in denen der Textausdruck enthalten ist. Ist der Textausdruck ein Leerstring (""), so werden sämtliche Zeilen angezeigt.
Der Zeilenbereich wird genau wie bei LIST angegeben. Im Unterschied zu LIST kann dieser Befehl jedoch innerhalb eines Programms verwendet werden, ohne daß ein Rücksprung in den Ready-Modus erfolgt.
Die folgende Zeile führt zur Ausgabe aller Programmzeilen zwischen 100 und 1000, in denen GOSUB 500 steht:

```
FIND "GOSUB 500",100-1000
```

Die Ausgabe kann mit ESC angehalten oder abgebrochen werden.

GPAPER <Farbstift>
KOMMANDO:
Wählt die Hintergrundfarbe für Textausgaben auf Grafikcursor-Position (nach TAG).

GPEN <Farbstift>, <Schreibmodus>
KOMMANDO:
Wählt die Schreibfarbe für Grafik und bestimmt, wie die neuen Bildpunkte mit der Hintergrundfarbe verknüpft werden sollen. Der zweite Parameter hat folgende Wirkung:
0 – normal
1 – XOR-Modus
2 – AND-Modus
3 – OR-Modus
Angewendet wird die Verknüpfung auf die Farbstiftnummer der Grafik-Schreibfarbe und des Hintergrundes.

```
10 TAG:MOVE 100,100
20 GPEN 2:GPAPER 3
30 PRINT "ABC";
```

MONITOR <Startadresse>, <Endadresse>, <RAM/ROM-Select>
KOMMANDO:
Gibt den Inhalt des angegebenen Speicherbereichs in hexadezimaler und ASCII-Form auf dem Bildschirm aus. Bei der ASCII-Darstellung wird bei Zeichen > 127 Bit 7 gelöscht. Die Steuerzeichen < 32 werden durch einen Punkt dargestellt. Der dritte Parameter ist wahlweise und hat folgende Bedeutung:
0 – RAM ausgewählt
1 – ROM ausgewählt
Wird er nicht angegeben, so wird der RAM-Bereich angezeigt. Die Ausgabe kann mit ESC angehalten oder abgebrochen werden.
Das folgende Kommando liest die Fehlermeldung des Basicinterpreters im oberen ROM:

```
MONITOR &CC5B,&CE66,1
```

PEN <#Window>, <1. Farbstift>, <2. Farbsstift>
KOMMANDO:
Wählt die Schreibfarbe für das angegebene Textfenster. Wird ein zweiter Farbstift angegeben, so werden die beiden Farben bei allen Textausgaben rasterförmig gemischt. Dadurch stehen insbesondere in MODE 1 effektiv mehr Farben zur Verfügung (maximal 10).

```
10 MODE 1:CHAR 12
20 INK 0,0:INK 1,6:INK 2,9:INK 3,13
30 FOR i=0 TO 3:FOR j=i TO 3
40 PEN i,j:PRINT "Bunte Mischung"
50 NEXT j,i
```

SET TIME <Stunden>, <Minuten>, <Sekunden>
KOMMANDO:
Dient zur Eingabe der Uhrzeit und stellt die XBASIC-Systemuhr.



10 SET TIME 10,30,0

TIME$

FUNKTION:

Gibt die Uhrzeit als Zeichenkette in der Form STUNDEN:MINUTEN:SEKUNDEN zurück. Diese Uhr ist unabhängig von dem normalen CPC-Timer und wird bei der Initialisierung von XBASIC mit 00:00:00 gestartet. Das folgende Programm verwandelt den CPC 464 in eine Digitaluhr:

```
10 MODE 1:INK 0,0:INK 1,12
20 CHAR 12
30 LOCATE 12,12:PRINT TIME$:GOTO 30
```

VAL(< Textausdruck >)

FUNKTION:

Gibt den Wert des Textausdrucks als numerischen Ausdruck zurück. Im Unterschied zur normalen VAL-Funktion des CPC wird der Text, falls nötig, auch als Formel interpretiert. Damit kann auf bequeme Weise die Eingabe von Funktionen für Mathematikprogramme oder Funktionenplotter per INPUT programmiert werden.

```
10 x=3
20 f$="sin(x/2)"
30 PRINT VAL(f$)
```

Die folgende, sehr leistungsfähige Gruppe von Befehlen dient dazu, auf dem CPC 464 externe Windows einzurichten und zu verwalten. Dem Inhalt dieser Bildschirmfenster wird ein separater Platz im RAM zugewiesen, wodurch sich ganz neue Möglichkeiten eröffnen:

- Kopieren von Bildschirmbereichen an eine andere Stelle
- Maximal drei voneinander unabhängige Bildschirmseiten
- Pull-Down-Windows: Fenster können auf den Bildschirm gerufen werden, ohne daß der bisherige Inhalt des Bereiches verloren geht.

SET WINDOW <Liste von Windowdaten>

KOMMANDO:

Definiert die Größe der maximal vier externen Fenster und reserviert den notwendigen Speicherplatz. Die Liste der Windowdaten enthält folgende Informationen:

Breite Window 0, Höhe Window 0, Breite Window 1, Höhe Window 1,...

Die Breite und Höhe der Fenster wird in Form von Textspalten bzw. Textzeilen angegeben. Die effektive Breite ist also vom aktuellen MODE abhängig. Das folgende Programm installiert zwei zusätzliche Bildschirmseiten:

```
10 MODE 1
20 SET WINDOW 40,25,40,25
```

RESET WINDOW

KOMMANDO:

Setzt alle durch SET WINDOW definierten Fenster zurück und gibt den Speicherplatz im RAM wieder frei.

Bitte beachten:

SET und RESET Window führen intern einen MEMORY-Befehl aus und verändern HIMEM. Deshalb kann es zu einem "Memory Full"-Error kommen, wenn der zur Verfügung stehende Speicherplatz nicht ausreicht oder gesperrt ist (z.B. durch einen Puffer für Kassetten I/O).

PUT WINDOW <Nummer>, <x-Koord.>, <y-Koord.>

KOMMANDO:

Legt einen Bildschirmbereich unter der angegebenen Nummer im Speicher ab. Die Größe wird durch SET WINDOW bestimmt, die linke obere Ecke durch die Textkoordinaten x,y. Wurde ein Fenster mit der Nummer noch nicht definiert, so folgt die Fehlermeldung "Improper Argument".

GET WINDOW <Nummer>, <x-Koord.>, <y-Koord.>

KOMMANDO:

Holt den Inhalt des externen Fensters mit der angegebenen Nummer auf den Bildschirm. Die Parameter haben die gleiche Bedeutung wie bei PUT WINDOW.

EXC WINDOW <Nummer>, <x-Koord.>, <y-Koord.>

KOMMANDO:

Tauscht einen Bildschirmbereich gegen den Inhalt eines externen Fensters aus ("Exchange"). Die Parameter haben die gleiche Bedeutung wie bei den vorhergehenden Befehlen.

Insbesondere dieser Befehl ermöglicht die Programmierung von Pull-Down-Windows, wie das Beispielprogramm 1 zeigt:

(Matthias Uphoff)

Beispielprogramm 1:

```
10 MODE 1:INK 0,0:INK 1,0:INK 2,0:INK 3,0
20 SET WINDOW 20,12,20,12,20,12
30 FOR W= 0 TO 2
40 PEN 0:PAPER W+1:CLS
50 LOCATE 7,6:PRINT"WINDOW";W
60 PUT WINDOW W,1,1
70 NEXT W
80 PEN 1:PAPER 0:CLS
90 INK 1,6:INK 2,12:INK 3,14
100 MOVE 320,200:GPEN 3
110 FOR R= 0 TO 150 STEP 10:CIRCLE R:NEXT
120 FOR W= 0 TO 2
130 EXC WINDOW W,(W+1)*5,(W+1)*4
140 FOR I = 1 TO 500:NEXT I,W
150 FOR W= 2 TO 0 STEP -1
160 EXC WINDOW W,(W+1)*5,(W+1)*4
170 FOR I = 1 TO 500:NEXT I,W
180 GOTO 120
```

Listing

```
10 '************************************      [1460]
20 'XBASIC-Erweiterung fuer den CPC 464       [3451]
30 '************ Basiclader ***********        [1738]
40 '                                          [117]
50 adr=&9DFF                                  [1000]
60 SYMBOL AFTER 256:MEMORY adr:SYMBOL AFTE    [2548]
R 240
70 FOR block=1 TO 16                          [1700]
80 s=0                                        [666]
90 FOR count=1 TO 128                         [1665]
100 READ a$:byte=VAL("&"+a$):s=s+byte         [2141]
110 adr=adr+1:POKE adr,byte                   [1847]
120 NEXT count                                [850]
130 READ check:IF s=check THEN PRINT"Block    [4859]
";block;"OK" ELSE PRINT "Datafehler in Blo
ck";block:END
140 NEXT block                                [382]
150 '                                         [117]
160 SAVE"XBASIC.BIN",b,&9E00,&800             [982]
170 CALL &9E00:END                            [994]
180 '                                         [117]
190 '********************** Block 1           [1026]
200 DATA 3E,C3,32,16,AC,21,3A,9E              [1261]
210 DATA 22,17,AC,32,19,AC,21,5D              [1324]
220 DATA 9E,22,1A,AC,32,07,AC,21              [747]
230 DATA 7E,9E,22,08,AC,32,0A,AC              [1835]
240 DATA 21,91,9E,22,0B,AC,32,0D              [775]
250 DATA AC,21,84,A5,22,0E,AC,21              [1475]
260 DATA D3,A5,01,32,00,50,59,C3              [790]
270 DATA E9,BC,11,A4,9E,CD,27,E3              [1456]
280 DATA D0,7E,CD,7B,FF,D8,F1,F1              [1016]
290 DATA 1A,D1,C1,F5,07,3E,E0,38              [1685]
300 DATA 02,3E,FF,CD,25,DF,F1,CD              [2350]
310 DATA 25,DF,C3,86,DF,79,07,30              [1505]
320 DATA 11,3E,E0,B9,C0,D1,C1,DD              [1705]
330 DATA E1,E1,7E,23,E5,DD,E5,C5              [1507]
340 DATA D5,4F,21,A4,9E,CD,13,E3              [1189]
350 DATA D0,7E,23,C1,C1,C9,FE,C0              [1652]
360 DATA 16272                                [299]
370 '********************** Block 2           [1033]
380 DATA C0,23,7E,87,D0,C1,4F,06              [979]
390 DATA 00,EB,21,26,9F,09,C3,BB              [1584]
```

```
400 DATA DD,D6,1E,FE,80,D0,D6,40          [1705]
410 DATA D8,C1,4F,06,00,E5,21,4C          [1175]
420 DATA 9F,C3,B4,D0,43,48,41,D2          [1106]
430 DATA 80,46,49,4C,CC,81,50,45          [1056]
440 DATA CE,82,47,50,45,CE,83,47          [480]
450 DATA 50,41,50,45,D2,84,43,49          [1569]
460 DATA 52,43,4C,C5,85,53,45,54          [1476]
470 DATA 20,54,49,4D,C5,8B,53,45          [713]
480 DATA D4,86,52,45,53,45,D4,87          [1065]
490 DATA 50,55,D4,88,47,45,D4,89          [954]
500 DATA 45,58,C3,8A,54,49,4D,45          [1517]
510 DATA A4,8C,45,44,D4,8D,4D,4F          [1547]
520 DATA 4E,49,54,4F,D2,8E,43,55          [844]
530 DATA 52,53,4F,D2,8F,46,49,4E          [2063]
540 DATA 14642                            [446]
550 '******************** Block 3         [1032]
560 DATA C4,90,44,45,43,A4,20,56          [1264]
570 DATA 41,CC,21,43,4F,50,59,A4          [1108]
580 DATA 22,00,63,2C,6E,78,74,64          [1230]
590 DATA 65,7A,0D,0A,35,39,31,35          [1358]
600 DATA 20,27,61,64,64,20,54,9F          [1792]
610 DATA 49,A0,BB,A1,15,A2,2A,A2          [1533]
620 DATA 30,A2,DB,A2,59,A3,7D,A3          [1530]
630 DATA 82,A3,87,A3,64,A5,7B,D0          [804]
640 DATA 90,A1,34,A4,28,A4,BE,A4          [646]
650 DATA 6C,0D,0A,35,EE,F8,2C,A1          [2298]
660 DATA 51,A1,64,20,3E,20,CD,FB          [1269]
670 DATA C1,32,E2,A5,11,0C,14,B7          [1296]
680 DATA 28,03,11,6A,9F,ED,53,DA          [1122]
690 DATA BD,C9,FE,20,DA,0C,14,CD          [1180]
700 DATA D3,12,11,08,A6,01,08,00          [571]
710 DATA 3A,E2,A5,CB,47,20,10,CB          [1578]
720 DATA 13758                            [355]
730 '******************** Block 4         [1039]
740 DATA 4F,28,18,41,7E,CB,3F,A6          [1696]
750 DATA 12,23,13,10,F7,18,0E,41          [1348]
760 DATA 7E,CB,3F,B6,12,23,13,10          [1153]
770 DATA F7,18,02,ED,B0,EB,3A,E2          [1637]
780 DATA A5,CB,67,28,04,2B,36,FF          [1295]
790 DATA 23,06,01,CB,57,28,27,DD          [978]
800 DATA 21,08,A6,06,08,C5,DD,4E          [882]
810 DATA 00,06,08,CB,09,DD,CB,00          [1682]
820 DATA 1E,DD,CB,10,1E,DD,CB,00          [1732]
830 DATA 2E,DD,CB,10,1E,10,EC,DD          [2342]
840 DATA 23,C1,10,E1,06,02,CB,5F          [1070]
850 DATA 28,18,C5,01,08,00,E5,09          [1268]
860 DATA EB,E1,41,2B,7E,1B,12,1B          [895]
870 DATA 12,10,F8,11,18,00,19,C1          [900]
880 DATA 10,E8,2A,96,B2,E5,21,08          [1104]
890 DATA A6,22,96,B2,3A,94,B2,F5          [1180]
900 DATA 12485                            [424]
910 '******************** Block 5         [1038]
920 DATA AF,32,94,B2,4F,CD,25,A0          [881]
930 DATA 79,CD,34,13,3A,E2,A5,CB          [1482]
940 DATA 57,28,09,0E,02,CD,25,A0          [1380]
950 DATA 79,CD,34,13,F1,32,94,B2          [929]
960 DATA E1,22,96,B2,C9,3A,E2,A5          [1760]
970 DATA CB,5F,C8,21,86,B2,E5,3A          [888]
980 DATA 89,B2,BE,28,08,3A,8B,B2          [1731]
990 DATA BE,38,02,2B,35,79,C5,CD          [1628]
1000 DATA 34,13,C1,E1,35,2B,34,0C         [1079]
1010 DATA C9,CD,4B,C2,E5,DF,52,A0         [2248]
1020 DATA E1,C9,55,A0,FE,CD,F6,17         [920]
1030 DATA F5,CD,FC,15,CD,27,18,CD         [1033]
1040 DATA 86,0C,C1,B8,C8,32,DF,A5         [1441]
1050 DATA CD,1A,16,CD,FF,16,D2,E2         [1456]
1060 DATA A0,E5,FD,E1,D5,DD,E1,CD         [1061]
1070 DATA A9,0B,3A,DF,A5,47,FD,23         [1092]
1080 DATA 17472                           [417]
1090 '******************** Block 6        [1045]
1100 DATA CD,2D,0C,3A,34,B3,FD,95         [1799]
1110 DATA 38,05,78,AE,A1,28,EF,FD         [1369]
1120 DATA 2B,CD,13,0C,51,59,E5,C5         [1521]
1130 DATA CB,01,DC,05,0C,78,42,AE         [984]
1140 DATA A1,57,20,0A,B0,28,07,DD         [1441]
1150 DATA 2B,DF,F2,A0,DD,23,C1,E1         [1658]
1160 DATA E5,C5,CB,09,DC,F9,0B,78         [1374]
1170 DATA 43,AE,A1,5F,20,0A,B0,28         [1368]
1180 DATA 07,DD,23,DF,F2,A0,DD,2B         [1293]
1190 DATA C1,E1,3A,38,B3,AE,A1,AE         [449]
1200 DATA 77,3A,36,B3,FD,95,30,0A         [818]
1210 DATA FD,2B,CD,13,0C,78,AE,A1         [1587]
1220 DATA 28,B4,2A,E0,A5,7D,B4,C8         [377]
1230 DATA 2B,22,E0,A5,DF,1A,A1,C3         [1520]
1240 DATA 6B,A0,F5,A0,FD,D5,2A,89         [2306]
1250 DATA AE,01,03,00,CD,18,F6,38         [1308]
1260 DATA 16228                           [295]
1270 '******************** Block 7        [1044]
1280 DATA 17,EB,22,89,AE,DD,E5,D1         [953]
```

```
1290 DATA FD,E5,C1,2B,72,2B,73,2B          [876]
1300 DATA 71,2A,E0,A5,23,22,E0,A5          [1246]
1310 DATA D1,C9,1D,A1,FD,2A,89,AE          [2338]
1320 DATA 2B,56,2B,5E,2B,22,89,AE          [1177]
1330 DATA 6E,26,00,C9,CD,70,D0,E5          [1100]
1340 DATA CD,35,A1,E1,C9,CD,DA,FB          [1110]
1350 DATA CA,DA,FF,EB,E5,16,00,58          [2487]
1360 DATA 19,5E,72,E3,D5,CD,BB,DE          [1521]
1370 DATA CD,FB,CE,D1,E1,73,C3,2D          [1813]
1380 DATA FF,CD,D0,C1,CD,27,C3,14          [1546]
1390 DATA 1C,D5,CD,55,DD,3E,01,DC          [1312]
1400 DATA 6D,CE,47,CD,37,DD,29,78          [1504]
1410 DATA CD,19,FC,E3,E5,CD,78,BB          [1468]
1420 DATA E3,C5,CD,87,BB,E5,CD,75          [1224]
1430 DATA BB,E1,CD,60,BB,38,02,3E          [1267]
1440 DATA 19121                            [377]
1450 '******************** Block 8         [1019]
1460 DATA 20,12,13,24,C1,10,EA,CD          [895]
1470 DATA BA,FB,E1,CD,75,BB,E1,C9          [1226]
1480 DATA CD,86,D6,CD,3C,FF,E5,EB          [1740]
1490 DATA E5,11,A4,AC,7E,B7,28,0B          [964]
1500 DATA 4F,06,00,23,7E,23,66,6F          [1416]
1510 DATA ED,B0,AF,12,CD,43,CA,CD          [2506]
1520 DATA DC,F7,CD,59,FB,D1,CD,66          [972]
1530 DATA FF,E1,C9,CD,12,C2,11,4A          [1065]
1540 DATA 13,ED,53,D4,BD,CD,55,DD          [1246]
1550 DATA D0,CD,4B,C2,CD,2C,BC,32          [1897]
1560 DATA E5,A5,11,DA,A1,ED,53,D4          [1443]
1570 DATA BD,C9,E5,EB,CD,D3,12,E5          [1159]
1580 DATA CD,11,A2,E1,06,08,0E,AA          [588]
1590 DATA 11,00,A6,D5,7E,A1,12,23          [1135]
1600 DATA 13,CB,01,10,F7,21,91,B2          [956]
1610 DATA 4E,36,9F,2E,8F,46,3A,E5          [1638]
1620 DATA 17806                            [309]
1630 '******************** Block 9         [1018]
1640 DATA A5,77,E1,D1,C5,CD,11,A2          [968]
1650 DATA C1,21,91,B2,71,2E,8F,70          [1297]
1660 DATA C9,D5,C3,4E,13,CD,4B,C2          [1602]
1670 DATA CD,DE,BB,CD,55,DD,D0,3E          [1531]
1680 DATA 04,CD,FB,C1,E5,CD,59,BC          [1153]
1690 DATA E1,C9,CD,4B,C2,C3,E4,BB          [1784]
1700 DATA 3E,F1,CD,FB,C1,4F,CD,55          [2032]
1710 DATA DD,DC,15,A2,E5,DF,42,A2          [1110]
1720 DATA E1,C9,45,A2,FE,CD,EC,0A          [906]
1730 DATA 2F,C6,01,CE,03,47,ED,44          [1008]
1740 DATA A1,4F,CD,FC,15,D5,E5,CD          [952]
1750 DATA 12,16,D5,E5,CD,57,16,CD          [581]
1760 DATA 04,16,6F,5F,E5,CD,50,37          [1072]
1770 DATA 22,E3,A5,E1,11,00,00,C5          [1502]
1780 DATA E5,D5,62,6B,CD,50,37,EB          [1225]
1790 DATA 2A,E3,A5,B7,ED,52,06,08          [1244]
1800 DATA 18733                            [289]
1810 '******************** Block 10        [1172]
1820 DATA 54,5D,D5,CB,3C,CB,1D,E5          [1603]
1830 DATA EB,CD,8C,37,D1,19,D1,10          [667]
1840 DATA F1,CB,3C,CB,1D,23,D1,7D          [1396]
1850 DATA E3,BD,30,02,2B,2B,D5,E5          [2211]
1860 DATA E5,D5,CD,16,18,D1,E1,CD          [751]
1870 DATA D4,37,E5,D5,CD,16,18,D1          [2352]
1880 DATA E1,EB,CD,D4,37,EB,D5,CD          [1957]
1890 DATA 16,18,D1,E1,E5,CD,16,18          [1669]
1900 DATA E1,D1,E3,7D,E3,BD,38,D4          [1426]
1910 DATA F1,C1,7B,80,5F,79,BB,30          [1995]
1920 DATA 9E,E1,D1,CD,04,16,E1,D1          [1111]
1930 DATA C3,F4,15,CD,37,DD,D8,CD          [1616]
1940 DATA 3E,FC,CD,71,A3,EB,2A,E6          [1702]
1950 DATA A5,7C,B5,20,06,2A,7B,AE          [1351]
1960 DATA 22,E6,A5,22,E8,A5,EB,CD          [1305]
1970 DATA 46,A3,E5,C5,C5,60,68,CD          [1460]
1980 DATA 19288                            [338]
1990 '******************** Block 11        [1175]
2000 DATA 43,A3,D5,26,00,6A,54,CD          [1390]
2010 DATA BE,BD,44,4D,2A,E8,A5,B7          [1300]
2020 DATA ED,42,22,E8,A5,54,5D,C5          [2070]
2030 DATA CD,F5,F4,C1,0B,23,E5,54          [1552]
2040 DATA 5D,13,36,00,ED,B0,E1,D1          [1812]
2050 DATA E3,29,29,01,F0,A5,09,72          [1594]
2060 DATA 23,73,23,D1,73,23,72,C1          [1043]
2070 DATA E1,CD,55,DD,D0,0C,CB,51          [1304]
2080 DATA 28,B5,C9,EF,95,0B,C5,CD          [621]
2090 DATA 27,C3,CD,17,BC,78,BA,DA          [584]
2100 DATA 05,C2,79,BB,DA,05,C2,C1          [921]
2110 DATA C9,CD,37,DD,D8,ED,5B,E6          [1471]
2120 DATA A5,7B,B2,C8,CD,3E,FC,CD          [1589]
2130 DATA F5,F4,11,00,00,ED,53,E6          [782]
2140 DATA A5,11,F0,A5,AF,06,10,12          [1252]
2150 DATA 13,10,FC,48,C9,01,C3,A3          [1156]
2160 DATA 17455                            [457]
2170 '******************** Block 12        [1170]
2180 DATA 18,08,01,D7,A3,18,03,01          [1703]
2190 DATA EB,A3,ED,43,C0,A3,CD,37          [239]
2200 DATA DD,D8,3E,04,CD,FB,C1,4F          [1525]
2210 DATA CD,37,DD,2C,CD,46,A3,E5          [1710]
2220 DATA C5,EB,CD,1A,BC,E3,26,00          [786]
2230 DATA 29,29,01,F0,A5,09,56,23          [590]
2240 DATA 5E,23,4E,23,46,E1,78,B1          [1493]
2250 DATA CA,05,C2,DF,C0,A3,E1,C9          [1196]
2260 DATA D7,A3,FF,D5,E5,7E,02,CD          [1862]
2270 DATA 02,A4,03,15,20,F7,E1,D1          [1461]
2280 DATA CD,0E,A4,1D,20,ED,C9,D5          [730]
2290 DATA E5,0A,77,CD,02,A4,03,15          [1070]
2300 DATA 20,F7,E1,D1,CD,0E,A4,1D          [1478]
2310 DATA 20,ED,C9,D5,E5,0A,5F,7E          [2348]
2320 DATA 02,73,CD,02,A4,03,15,20          [1773]
2330 DATA F4,E1,D1,CD,0E,A4,1D,20          [1251]
2340 DATA 16245                            [278]
2350 '******************** Block 13        [1173]
2360 DATA EA,C9,2C,C0,24,7C,E6,07          [848]
2370 DATA C0,7C,D6,08,67,C9,7C,C6          [788]
2380 DATA 08,67,E6,38,C0,7C,D6,40          [1283]
2390 DATA 67,7D,C6,50,6F,D0,24,7C          [1020]
2400 DATA E6,07,C0,7C,D6,08,67,C9          [1647]
2410 DATA 3E,02,CD,FB,C1,B7,CA,84          [1521]
2420 DATA BB,C3,81,BB,CD,91,CE,D5          [1145]
2430 DATA CD,37,DD,2C,CD,91,CE,CD          [1492]
2440 DATA 55,DD,3E,02,DC,FB,C1,E6          [1574]
2450 DATA 01,28,02,3E,03,2F,32,BB          [1883]
2460 DATA A4,CD,8D,C3,E3,EB,B7,ED          [938]
2470 DATA 52,DA,05,C2,EB,7C,CD,9F          [1342]
2480 DATA A4,7D,CD,9F,A4,CD,B4,A4          [407]
2490 DATA CD,B4,A4,E5,06,08,DF,B9          [1186]
2500 DATA A4,CD,9F,A4,CD,B4,A4,23          [1113]
2510 DATA 10,F4,CD,B4,A4,E1,06,08          [1450]
2520 DATA 18625                            [297]
2530 '******************** Block 14        [1184]
2540 DATA DF,B9,A4,CB,BF,FE,20,30          [1312]
2550 DATA 02,3E,2E,CD,5A,BB,23,1B          [1197]
2560 DATA 10,EE,CD,92,C3,CD,3C,C4          [1472]
2570 DATA 7A,B7,F2,5D,A4,E1,C9,F5          [1233]
2580 DATA 0F,0F,0F,0F,CD,A8,A4,F1          [2231]
2590 DATA E6,0F,FE,0A,38,02,C6,07          [1180]
2600 DATA C6,30,18,02,3E,20,C3,5A          [1927]
2610 DATA BB,BC,A4,FF,7E,C9,CD,9F          [1295]
2620 DATA CE,ED,53,EA,A5,32,EC,A5          [1754]
2630 DATA CD,29,A5,CD,C6,C1,F5,CD          [1219]
2640 DATA 4A,DD,E5,CD,DB,A4,E1,F1          [1816]
2650 DATA C3,A2,C1,D5,50,59,CD,A3          [1540]
2660 DATA E7,D1,4E,23,46,2B,78,B1          [1138]
2670 DATA C8,CD,3C,C4,E5,09,E3,D5          [1943]
2680 DATA E5,23,23,5E,23,56,E1,E3          [1242]
2690 DATA CD,B8,FF,E3,38,28,CD,63          [1590]
2700 DATA 18686                            [292]
2710 '******************** Block 15        [1187]
2720 DATA E1,E5,79,95,3C,47,EB,2A          [1129]
2730 DATA EA,A5,3A,EC,A5,4F,3E,01          [1048]
2740 DATA CD,DC,FA,E1,B7,28,0B,CD          [1329]
2750 DATA 45,E1,23,7E,B7,20,F8,CD          [1345]
2760 DATA 4E,C3,D1,E1,18,BC,E1,E1          [1056]
2770 DATA C9,01,01,00,11,FF,FF,CD          [1560]
2780 DATA 55,DD,D0,FE,23,C8,FE,F5          [985]
2790 DATA 28,0A,CD,58,A5,42,4B,C8          [933]
2800 DATA CD,55,DD,D8,CD,37,DD,F5          [1515]
2810 DATA 11,FF,FF,C8,CD,55,DD,D8          [978]
2820 DATA 7E,CD,58,A5,C4,55,DD,C9          [1517]
2830 DATA CD,2C,D0,F5,CD,45,FF,D2          [1555]
2840 DATA 7B,D0,F1,C9,3E,18,CD,FB          [1690]
2850 DATA C1,47,CD,7B,A5,4F,CD,7B          [1667]
2860 DATA A5,F3,32,ED,A5,ED,43,EE          [1205]
2870 DATA A5,FB,C9,CD,37,DD,2C,3E          [966]
2880 DATA 19969                            [265]
2890 '******************** Block 16        [1182]
2900 DATA 3C,C3,FB,C1,FE,8C,C0,2B          [1491]
2910 DATA 7E,FE,E0,C0,F1,23,E5,3E          [1617]
2920 DATA 08,CD,19,FC,EB,11,EF,A5          [414]
2930 DATA 06,03,1A,0E,2F,0C,D6,0A          [1225]
2940 DATA 30,FB,C6,3A,71,23,77,05          [1322]
2950 DATA 28,07,23,36,3A,23,1B,18          [1107]
2960 DATA E9,CD,BA,FB,E1,C3,3F,DD          [465]
2970 DATA 21,ED,A5,34,7E,FE,3C,D8          [883]
2980 DATA 36,00,23,34,7E,FE,3C,D8          [1089]
2990 DATA 36,00,23,34,7E,FE,18,D8          [1018]
3000 DATA 36,00,C9,20,27,6A,70,20          [961]
3010 DATA 26,63,31,00,81,B8,A5,00          [1361]
3020 DATA 00,00,00,00,00,00,00,00          [1005]
3030 DATA 00,00,00,00,00,00,00,00          [1005]
3040 DATA 00,00,00,00,00,00,00,00          [1005]
3050 DATA 00,00,00,00,00,00,00,00          [1005]
3060 DATA 10796                            [397]
```

CompuCamp ein voller Erfolg!

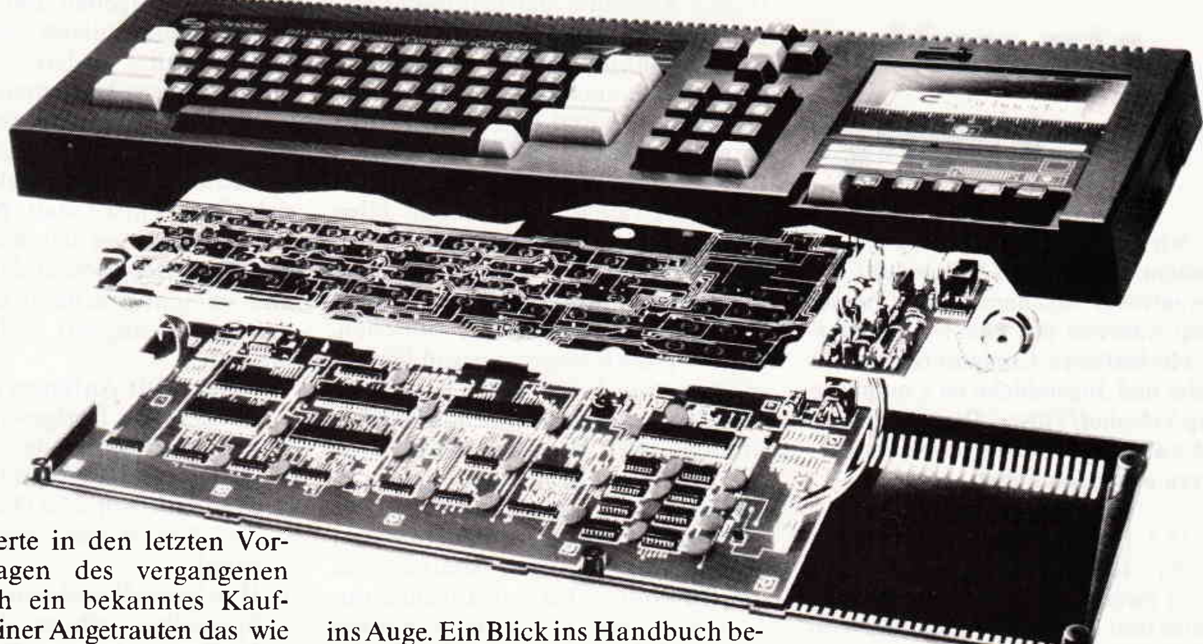
Wie wir in den vorigen Ausgaben berichteten, veranstaltet Schneider CPC International zusammen mit CompuCamp während der Oster-, Sommer- und Herbstferien Computercamps für Kinder und Jugendliche im Computercamp Veltishof/Titisee. Das erste Camp fand während der Osterferien statt. Hierzu unser Rückblick.

Am 15.3. 1986 startete das Ostercamp in der Jugendherberge Veltishof. Schon zwei Tage vorher wurden bei Schnee und Eis die Vorbereitungen für die folgenden drei Wochen getroffen. Am Samstag reisten dann die 17 Teilnehmer der ersten Woche an. Sonntag vormittag begannen dann die Kurse Basic I und Basic II mit dem Feststellen des Wissensstandes und der Interessengebiete. Das Freizeitprogramm startete mit einem Spaziergang über den zugefrorenen Titisee zur Ortschaft, um die Umgebung zu erkunden. Der Abend wurde beim "freien Programmieren" am Computer verbracht und natürlich wurden auch Computerspiele gespielt.
Während man sich am nächsten Tag in den Kursen mit For-Next-Schleifen und 3D-Balkendiagrammen beschäftigte, machte sich die jeweils unterrichtsfreie Gruppe mit der herbergseigenen Skiausrüstung auf die Loipe direkt hinter der Herberge. Dort mußten einige einsehen, daß der Weg zum Profi doch sehr weit ist.
Am Mittwoch folgte dann die Krönung der Woche: ein Tagesausflug mit dem Bus durch den Schwarzwald. Er begann mit einer kurzen aber dafür sehr interessanten Besichtigung der Kuckucksuhren-Fabrik Hoenes in Neustadt, die wohl einige der schönsten Kuckucksuhren der Welt herstellt. Besonders bestaunt wurden die kunstvollen Schnitzereien, die in erstaunlich kurzer Zeit von Hand angefertigt werden, wie uns erzählt wurde. Es folgte ein Besuch der Glashütte in Wolfach, in der mundgeblasene Gläser hergestellt und von Hand geschliffen werden.
Nach einer längeren Rückfahrt erreichten wir die letzte Station der Fahrt, das Uhrenmuseum in Furtwangen. Die scheinbar auf unser besonderes Interesse abgestimmte Führung befaßte sich vor allem mit den Automatenuhren, die eigentlich die frühen Vorläufer der Computer waren. Gezeigt wurden nicht nur die Kuckucksuhren und ihre Verwandten, sondern auch wahre Prunkstücke mechanischer und holzbearbeitender Kunst, wie ein automatisches Klavier, verschiedene Pfeifenorgeln und eine durch Gewichte angetriebene Jahrtausenduhr mit Jahresanzeige, die auf einer Walze das Zeitprogramm von ca. 10.000 Jahren gespeichert hat. Spätestens jetzt mußte jeder einsehen, daß Programmsteuerung und Datenspeicherung keineswegs Errungenschaften unseres Jahrhunderts sind, sondern daß schon Jahrhunderte vor uns Menschen Programme geschrieben haben, die dann von Maschinen ausgeführt wurden. Außerdem waren Uhren aller Bau- und Abarten, von der Kirchturm- bis zur Taschenuhr, zu bewundern. Diese letzte Station war wohl der gelungenste Punkt des Tages, nicht zuletzt wegen der spritzigen Art des Vortrages und dem starken technischen Interesse der Gruppe. Der Abend wurde wieder beim "freien Programmieren" am Computer verbracht, aber ein Uhrenprogramm hat unseres Wissens nach keiner geschrieben! (?)
Am Donnerstag hatten die Wasserratten Gelegenheit ins Hallenbad zu gehen, verbunden mit einer Einkaufsmöglichkeit in Neustadt, und nach dem Abendessen war Video-Time.
Der letzte Tag war von starkem Schneetreiben gekennzeichnet. Was also blieb übrig, als die Computer-Spiele-Kiste aufzuklappen und die Videorekorder warmlaufen zu lassen. Logisch, daß auch die Computer auf vollen Touren liefen.
Abschied nehmen hieß es dann am Samstag. Die erste Gruppe wurde von den neuen Teilnehmern abgelöst.
Die folgenden Wochen waren genauso reizvoll wie die erste. Neben einem Ausflug zum Bodensee auf die Insel Mainau und einem Besuch bei Fischertechnik in Tumlingen gab es noch viele interessante Aktivitäten.
Diese ersten Wochen zeigten uns ganz deutlich, daß dieser neue Veranstaltungsort seine ganz besonderen Reize hat und sicher auf großes Interesse stößt. Gerade die sehr schöne Umgebung verspricht im Sommer ein besonderes Erlebnis zu werden.
Es zeigte sich außerdem, daß das Interesse am Schneider CPC noch größer ist, als zunächst erwartet und daher unser neues Angebot, speziell Maschinensprache mit dem Z-80-Prozessor und Hardwarebasteleien, vielen sehr gelegen kommen wird.
Die Jugendherberge als Veranstaltungsort für Computercamps hat sich in den vergangenen drei Wochen bewährt, nicht zuletzt aufgrund der Möglichkeit, andere gleichaltrige Herbergsgäste kennenzulernen.
Auch in den Sommerferien finden die Computercamps nach bewährtem Konzept im Computercamp Veltishof am Titisee statt. Nachfolgend eine Aufstellung der Kurse, welche größtenteils aufeinander aufbauen und dementsprechend einzeln oder aufeinanderfolgend belegt werden können.

1. Basic I (für Anfänger)
2. Basic II (für Fortgeschrittene)
3. Pascal (Einführung in die hochstrukturierte Programmiersprache)
4. Maschinensprache (Einführung in die Z-80-Assemblerprogrammierung)
5. Hardware-Basteleien (in diesem Spezialkurs gibt es alles, was das Bastlerherz begehrt. Kernstück des Kurses werden die Baukästen von Fischertechnik-computing sein)
6. Profi-Anwendungen (professionelle Anwendungen werden beleuchtet und Berufsperspektiven aufgezeigt)

Jedem Teilnehmer steht sein "eigener" Schneider CPC mit Peripherie zur Verfügung. Die Betreuung während des Campaufenthaltes ist durch geschulte Teams von Computerlehrern und Freizeitpädagogen gewährleistet. Redaktionsmitglieder von Schneider CPC International werden in regelmäßigen Abständen den Fragen der "Computercamper" Rede und Antwort stehen.
Das Computercamp liegt mitten im Schwarzwald in unmittelbarer Nähe des Titisees. Die wunderschön gelegene Jugendherberge garantiert höchsten Erholungswert. Für eine aktive Freizeitgestaltung bieten sich zahlreiche Einrichtungen an.
Das Fernsehen berichtete übrigens am 27.4. 1986 im ARD-Ratgeber-Reisen über die CompuCamp-Computercamps.
Die Nachfrage nach den Computercamps in den Sommerferien ist sehr, sehr groß. Aufgrund der bewußt begrenzten Teilnehmerzahl gibt es nur noch wenige freie Plätze. Wer noch einen dieser Plätze ergattern möchte, sollte noch heute den im Heft befindlichen Coupon als Voranmeldung ausfüllen und an CompuCamp, Goßlerstraße 21, 2000 Hamburg 55 schicken. Am besten ist es, vorher unter der Telefon-Nr. (040) 86 12 55 anzurufen und zu fragen, ob zu dem gewünschten Zeitpunkt noch Plätze frei sind.

SCHNEIDERWARE #1



Ich schlenderte in den letzten Vorweihnachtstagen des vergangenen Jahres durch ein bekanntes Kaufhaus, um meiner Angetrauten das wie immer verspätete Weihnachtsgeschenk auszusuchen. Während meines Rundganges kam ich wie von einer unsichtbaren Hand gezogen an der Computertheke vorbei. Kleine und große Kinder drängelten sich um die Eingabetastatur eines renommierten Computers. Sie ließen die Maus über den Tisch flitzen und erfreuten sich der aufwendigen Technik dieses kleinen Nagers. Gleich daneben stand ein vielversprechendes Gerät eines englischen Herstellers. Von der Neugier getrieben, sprach ich einen etwa 10-jährigen Jungen an, der sich wohl wegen seiner Größe bei den älteren Jungs nicht so ganz durchsetzen konnte. Ich fragte ihn, wie ihm denn der SCHNEIDER gefiele (den ich selbst in meinem Haus zum Ärger meiner Frau seit drei Wochen hegte und pflegte.

Die einzige Antwort, die er mir in echt bayerischer Manier gab, war folgende: „Do gibds doch no net a mol wos zum kaafen, a Maus homs a kaane". Völlig überrascht von seiner spontanen Antwort, begab ich mich zu meinem fahrbaren Untersatz und trat den Heimweg an.

Diese kleine Geschichte trug sich einige Wochen nach Erscheinen des "Jüngsten" der Schneiderfamilie in einer fränkischen Kleinstadt zu. Damals war noch nicht abzusehen, welch einen ungeheuren Siegeszug dieser Name "SCHNEIDER COMPUTER DIVISION" machen würde. Heute sind wir eines Besseren belehrt. Bis zum heutigen Tag befinden sich vier dieser stattlichen Modelle auf dem Markt. Wenn man diese großartigen Geräte einmal näher betrachtet, dann fallen einem schon einige Merkmale ins Auge. Ein Blick ins Handbuch bestätigt diese Beobachtung; nämlich das Vorhandensein zahlreicher Schnittstellen, schön versteckt an der Rückseite des Keyboards. Mit diesen Toren zur Außenwelt wollen wir uns in den nächsten Beiträgen intensiv befassen.

Den Notstand, den der Junge aus dem Kaufhaus so nett umschrieben hatte, möchten wir mit unserer Serie "SCHNEIDERWARE" endgültig beseitigen. Besondere Leckerbissen dieser Serie werden ein komplettes System zur Aus- und Eingabe von Daten, zur Steuerung externer Geräte (Drucker, Plotter), Programmieren von Eproms, eine ROM-Erweiterung, Vordergrund-ROMs sowie ein V/24-Interface mit den entsprechenden Softwarepaketen sein. Diese im Europakartenformat entwickelten Karten können einzeln oder als Gesamtsystem über einen gebufferten Bus im 19"-Gehäuse (mit ECB-Anschlußbelegung) am CPC betrieben werden. In weiteren Ausbaustufen wird sich zu den einfacheren Karten auch noch eine sogenannte Profireihe "Meßlabor zum Messen von elektrischen Größen" dazugesellen. Das Ganze wird natürlich noch durch ein leistungsstarkes externes Netzteil abgerundet. Anfragen an die Redaktion bestätigten uns in der Annahme, daß tatsächlich genügend Einsteiger damit beschäftigt sind, ihre Hardwareerweiterungen selbst zu stricken. Unsere Beiträge sollen demjenigen helfen, der gerne viel mehr mit seinem geliebten Stück anfangen möchte, als nur Programme von den unzähligen Zeitschriften abzutippen und sich hinterher zu ärgern, daß wieder mal ein Programm nicht läuft, weil in einer Datazeile ein Komma vergessen wurde. Wie ich eingangs schon erwähnte, ist der Schneider mit zahlreichen Schnittstellen gesegnet. Wir wollen uns den Begriff "Schnittstelle" etwas näher ansehen. Die eingebauten Schnittstellen der CPC's werden wir anhand der vier Modelle etwas näher betrachten.

Die Schnittstelle – das Tor zur Außenwelt

Die Überschrift sagt eigentlich schon alles über eine Schnittstelle aus. Eine derartige Einrichtung besitzt eigentlich jeder Computer. Man kann hier zwischen zwei Arten von Schnittstellen unterscheiden. Es gibt Soft- und Hardwareschnittstellen. Erstere bilden eine Verbindung zu Programmen oder deren Teilen innerhalb eines Systems. Man könnte sagen, die Möglichkeit, einen Programmablauf von außen zu verändern, wäre solch eine Softwareschnittstelle (Vektortechnik im 464). Uns beschäftigen mehr die Hardwareschnittstellen der CPC-Modelle. Deren gibt es beim 464 drei Wichtige; beim 664 und 6128 sind es schon zwei mehr. Durch das eingebaute Floppylaufwerk ist der Anschluß einer Zweitfloppy auf einem separaten Port herausgeführt. Die wohl wichtigste Schnittstelle ist der 50 Pol.Expansionsport neben dem unvollständigen Centronicsport. Die Anschlußbelegung ist bei allen drei Typen gleich, was für den Anwender ein großer Vorteil ist.

Der JOYCE besitzt ebenfalls einen Erweiterungsbus. Dieser ist aber in der Signalbelegung und -Bezeichnung nicht kompatibel zu den CPC's. Einige Signale fehlen ganz, so z.B. die Freigabe- und Sperrsignale für internen Speicher. Da das Konzept des JOYCE ein ganz anderes ist, müssen wir uns bei der Beschreibung auf die drei gleichartigen Typen beschränken.
Hardware ist nicht nur etwas für hochstudierte Leute. Früher war das vielleicht einmal so, aber seit die Integration der Peripheriebausteine derart fortgeschritten ist, können sich auch Leute wie Sie und ich mit dieser Thematik beschäftigen. Man muß nur noch wissen, was aus diesen vielbeinigen Käfern herauskommt, wenn man etwas Bestimmtes hineinsteckt. Von diesen integrierten Schaltkreisen behaupten auch in Ihrem Schneider einige ihren Platz. Da gibt es programmierbare Parallelinterfaces (PPI), 8-Bit-Latches (CENTRONICS) und einen hochintegrierten Krachmacher, den Soundchip. Die Ausführung, die der Schneider besitzt, hat noch eine Reihe anderer wichtiger Aufgaben, als nur Krach zu stiften. Er besitzt einen bidirektionalen Ein-/Ausgabeport, der die Aufgabe hat, die Tastatur und den Joystickport abzutasten.
Der Meister unter den Schnittstellen im CPC ist der 8255, der aus der Prozessorfamilie 8080 stammt. Er koordiniert innerhalb der CPC's alle Aktivitäten mit der Außenwelt. Er steuert die Kassetten-Schnittstelle, gibt Daten an den Soundchip (Tastaturabfrage) und verarbeitet ankommende Steuersignale wie z.B. das BUSY vom Centronicsport.

Ein weiterer einfacher Baustein befindet sich bei diesem Centronicsport. Dieses IC ist ein 8-fach-D-Flip-Flop des Typs 74 273. Mit diesem Baustein kann man sehr einfach ein Ausgabe-Interface zurechtschneidern, welches uns 8 Bit liefert. Solche hochtrabenden Dinge werden wir aber erst später tun. Zunächst müssen wir uns noch den Anschluß, der zweifelsfrei der wichtigste ist, etwas näher ansehen. Der Expansionsport ist, wie der englische Name schon sagt, ein Port, mit dem man Erweitern kann. Dieser Anschluß stellt uns fast alle Signale, die der Prozessor Z80 im Inneren der CPC's erzeugt, zur Verfügung. Das angesteckte Floppylaufwerk z.B. benötigt beim 464 einen Großteil dieser Leitungen. Die beiden anderen Rechner verarbeiten diese Signale schon im Inneren, was uns aber nicht weiter interessieren soll.

Der Autor stellt sich vor:



Mein Name ist Peter Richter. Vor ca. zehn Jahren fing ich an, auf dem damals noch "sehr" teuren Computeraltvater ZX 81 herumzuklopfen. Der kleine Speicherbereich trieb mich schon sehr bald ans Entwickeln von Speichererweiterungen. Aufgebohrt bis an die Grenze des Machbaren, stellte ich nach Erscheinen meines CPC's den guten alten Sinclair in den Schrank, wo er heute noch sein Dasein fristet. Mittlerweile bin ich 33 Jahre alt und noch kein bißchen weise. Auch heute beschäftige ich mich zum Leidwesen meiner Familie ausschließlich mit meinem CPC. Eine besondere Leidenschaft von mir ist es, jegliche Hardware auf Herz und Nieren zu erkunden. Da man vom Hardwareanalysieren nicht leben kann, verdiene ich meinen Lebensunterhalt als Außendienstmitarbeiter in einem Dienstleistungsbetrieb. In dieser Eigenschaft beschäftige ich mich mit der Wartung und Reparatur von Datenverarbeitungsanlagen. Ich habe mir einige Leckerbissen zum Thema "Hardware" einfallen lassen. Ich hoffe, daß sich durchaus brauchbare Anwendungen darunter befinden.

16 + 8 + 4: Adressen, Daten und Steuerung

Die wichtigsten "Strippen", die wir für unsere Erweiterungen benötigen, sind 16 Adress- und 8 Datenleitungen. Die Belegung dieser Leitungen können Sie Ihrem Handbuch entnehmen. Sie werden die Bezeichnungen A0-A15 und D0-D7 sehr leicht wiederfinden.
Wie Sie sicher wissen, ist die kleinste Informationseinheit das BIT (BInary digiT). Diese und die nächstgrößeren Einheiten "Nibble" (4 Bit) und "Byte" (8 Bit) werden uns während unserer ganzen Hardwarebastelei verfolgen.

Um den Begriff BIT zu verstehen, betrachten wir einmal eine Klingelanlage. Drücken wir auf den Knopf, so dröhnt die Glocke los (entspr. "High" = 5 V). Lassen wir nun den Klingelknopf wieder los, so entspricht das einem "Low" oder 0 Volt, wenn man es in Spannungspegeln ausdrückt. Um solche Vorgänge anschaulicher zu machen, drückt man in der Elektronik solche Zusammenhänge in sogenannten Zustandsdiagrammen aus.
Auf dem Datenbus, repräsentiert durch die 8 Datenbits D0-7, werden solche Spannungssprünge in schneller zeitlicher Folge übertragen. Die Datenleitungen in einem Microsystem werden immer parallel geschaltet, d.h. alle Komponenten, die mit Daten versorgt werden müssen, besitzen mehrere Datenleitungen, über welche gleichzeitig Daten übertragen werden. Solche Gruppen von Leitungen nennt man Bussystem. Dieses wird unterteilt in drei Gruppen: Datenbus, Adressbus und Steuer-/Kontrollbus. Die Zusammenarbeit dieser drei Gruppen läßt sich anschaulich anhand von Speicherzugriffen erklären. Wenn man einem Speicherbaustein an seinen Datenleitungen Informationen anbietet, so wird er diese Spannungspegel nicht Byte für Byte sinnlos in sich hineinschaufeln, sondern jedes Byte wohlgeordnet an seinem Platz aufbewahren. Für diese Organisationsaufgabe benötigt der Speicher die Adressleitungen. Mit Hilfe dieser Leitungen kann er, unterstützt von seiner internen Auswahllogik, jede Speicherzelle gezielt beschreiben und auslesen. Diese Tätigkeit wird "Adressieren" genannt. Um diesen Vorgang zu erklären, benutzen wir wieder einen Begriff der Digitaltechnik, die "Wahrheitstabelle". Mit solch einer Tabelle kann man die Ausgangszustände "logischer" Bausteine in Abhängigkeit ihrer Eingangszustände übersichtlich darstellen. Wir erstellen uns eine Tabelle mit drei Eingangsleitungen (A0-A2). Mit diesen drei Leitungen werden sich genau sieben mögliche Kombinationen einstellen, wie Tabelle 1 zeigt. Die Anzahl der

Tabelle 1

A2	A1	A0	DEZIMAL
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
4	2	1	WERTIGKEIT

Adressleitungen, die man benötigt, um eine bestimmte Anzahl Speicherzellen gezielt adressieren zu können, kann man nach folgender Formel berechnen: 2 hoch die Anzahl der verfügbaren Leitungen. Hat man nun vier Adressleitungen zur Verfügung, so kann mit ihnen schon der doppelte Speicherplatz adressiert werden (2 hoch 4 = 16). Mit den uns zur Verfügung stehenden 16 Adressleitungen können wir insgesamt 2 hoch 16 = 65536 Bytes = 64 KB Speicherplatz ansprechen.
Statische Speicher gibt es heute bis zu einer Größe von acht Kilobyte, das entspricht 8 * 8 * 1024 einzelnen Speicherzellen (=65536 Bit).
Wegen ihrer 8-Bit-Struktur gibt man nicht die Anzahl der einzelnen Speicherzellen an, sondern den achten Teil davon. Daß diese Datenübertragung mit einer sehr großen Geschwindigkeit geschehen muß, versteht sich eigentlich von selbst, wenn Sie bedenken, welch ein Aufwand getrieben werden muß, um dem Speicher zu sagen, er soll das Byte mit dem Wert &FF in die Speicherzelle &028F einschreiben. Da diese RAM (Random Access Memory)-Bausteine nicht nur Daten in ihr Inneres schreiben, sondern diese Daten auch wieder auslesen müssen, braucht man noch zwei andere Signale, die diese Aktivitäten unterscheiden können. Diese Signale bezeichnet man als READ (lesen) und WRITE (schreiben), wobei die entsprechende Tätigkeit durch LOW-Pegel des jeweiligen Signals ausgelöst wird. Man nennt diese Signale "Active-Low" und bezeichnet sie in der Kurzschreibweise mit einem darübergestellten Querstrich. Die nicht aktivierten Speicher verhalten sich dann so, als wären sie nicht angeschlossen. Sie verharren in einem hochohmigen Zustand. Diese Betriebsart kann man mit weiteren Signalen einstellen. Bei Speichern heißen solche Signale meist "OE" (Output Enable) oder "OD" (Output Disable). Ein Anschluß, der noch nicht erwähnt wurde, ist der "CS" (Chip Select)-Pin. Dieser hat die Aufgabe, wenn er auf LOW liegt, den Speicher aus seiner Schlafstellung aufzurütteln und ihm mitzuteilen, daß irgend jemand etwas von ihm will. Alle Aktivitäten in einem Computersystem laufen somit wohlgeordnet in einer genauen zeitlichen Reihenfolge ab. Nachdem sich in unserem Computer auch Ein-/Ausgabeports befinden, muß der Z80 auch noch wissen, welchen er zu welchem Zeitpunkt bedienen soll. Zu diesem Zweck stellt der Prozessor noch zwei wichtige Leitungen zur Verfügung, die da heißen "IOREQ" und "MREQ" (Input/Output REQuest und Memory REQuest). Liegen nun an diesen Pins Signale (es kann immer nur eines aktiv sein) mit LOW-Pegel an, so bedeutet dies, daß die Daten am Datenbus für eine Adresse gültig sind. Das angesprochene Port-IC kann nun die Daten verarbeiten und z.B. zum Drucker schicken. Beim Schreiben in den Speicher läuft dieser Vorgang genauso ab, nur daß hier das Signal "MREQ" eine Speicheroperation anzeigt. In unserem System gibt es natürlich noch eine Menge weiterer Signale, die aber für unsere Betrachtungen im Moment noch nicht so bedeutend sind, denn uns genügt zunächst das Wissen über den Daten- und Adressbus, um einen kleinen Einblick in ein Computersystem zu bekommen. Bild 1 zeigt Ihnen die grobe Struktur eines "Minimalsystems". Deutlich erkennt man die Verbindungen zu den einzelnen Komponenten wie CPU, Speicher und Ausgabeeinheiten, dargestellt durch die Pfeile.
Hier erkennt man, daß der englische Begriff "BUS" eigentlich nur bedeutet, daß über mehrere Leitungen parallel, das heißt mehrere Informationen zur gleichen Zeit von einem Ort zum anderen transportiert werden. Der Vorteil dieser Busstruktur ist der, daß sehr viele Informationen mit großer Geschwindigkeit übertragen werden können. Dieses Vorteils bedient man sich auch bei externen Erweiterungen.
Hier kann man durch diese Busstruktur, ansteckbaren Erweiterungskarten immer die gleichen Anschlüsse verpassen, und findet so stets gleiche Verhältnisse vor. Ferner kann man die einzelnen Karten untereinander austauschen.
Der einzige Nachteil bei dieser parallelen Verarbeitung ist die Tatsache, daß die Leitungen nicht unendlich lang gemacht werden können. Der Grund dafür ist in der hohen Verarbeitungsgeschwindigkeit zu suchen, denn hier treten einige Probleme auf, die uns aber nur wenig berühren, solange unsere Leitungen nicht länger als 50 cm sind. Unsere Erweiterungskarten werden sich in solch einem Bussystem befinden. Selbstverständlich können die Karten auch solo am Schneider betrieben werden. Die Karten werden so entwickelt, daß sie an allen drei CPC's lauffähig sind. Sollte das aus irgendeinem Grunde nicht möglich sein, so wird in dem entsprechenden Artikel darauf hingewiesen. Wir werden die Karten in ein 19"-Gehäuse integrieren, das den Vorteil großer Stabilität bietet, vom profimäßigen Aussehen ganz zu schweigen. Die modulare Struktur bietet Ihnen die Möglichkeit, sich schrittweise in die Problematik der Baugruppen einzuarbeiten.
Die Adressdecodierung ist genaugenommen nicht die interne Decodierung dieses Speicherbereiches, sondern vielmehr die Einordnung eines mehr oder minder großen Blockes in den Speicherbereich. Jedes IC hat außer den Daten- und Adressleitungen eine oder mehrere Freigabeleitungen. Die interne Adressierung funktioniert nur dann, wenn ein Freigabesignal am IC liegt und genau in diesem Moment Daten am Datenbus verfügbar sind. Um ein Freigabesignal zu erhalten, verknüpft man die hochwertigen Adressleitungen zu Steuersignalen, die für den zeitlichen Ablauf der Speicher- oder Ausgabeoperation unerläßlich sind. Diese Freigabesignale, die aus den höherwertigen Adressbits über sogenannte Adressdecoder (z.B. 74 LS 138) erzeugt werden, sind mit dem Kürzel "CS" (ChipSelect) bezeichnet.

Die Auswahl leicht gemacht

In der Microcomputertechnik sind

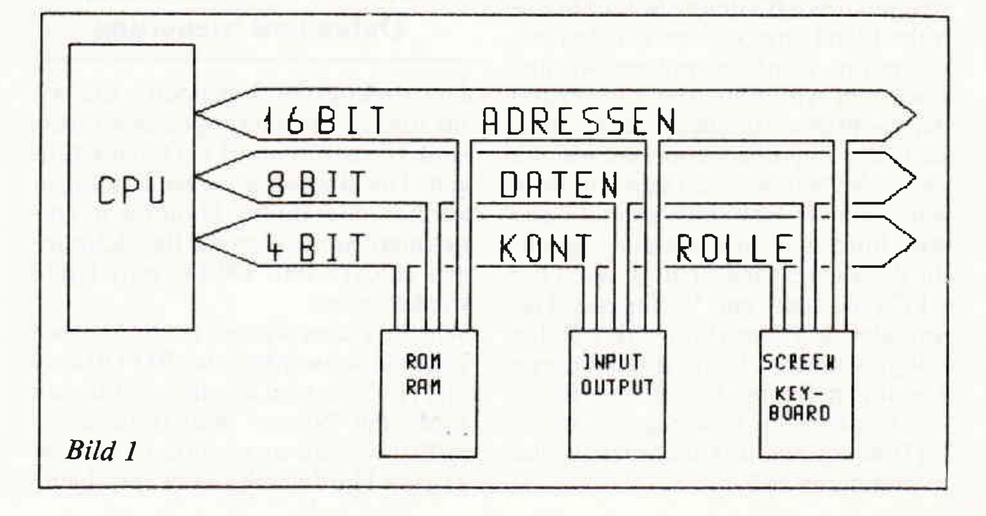
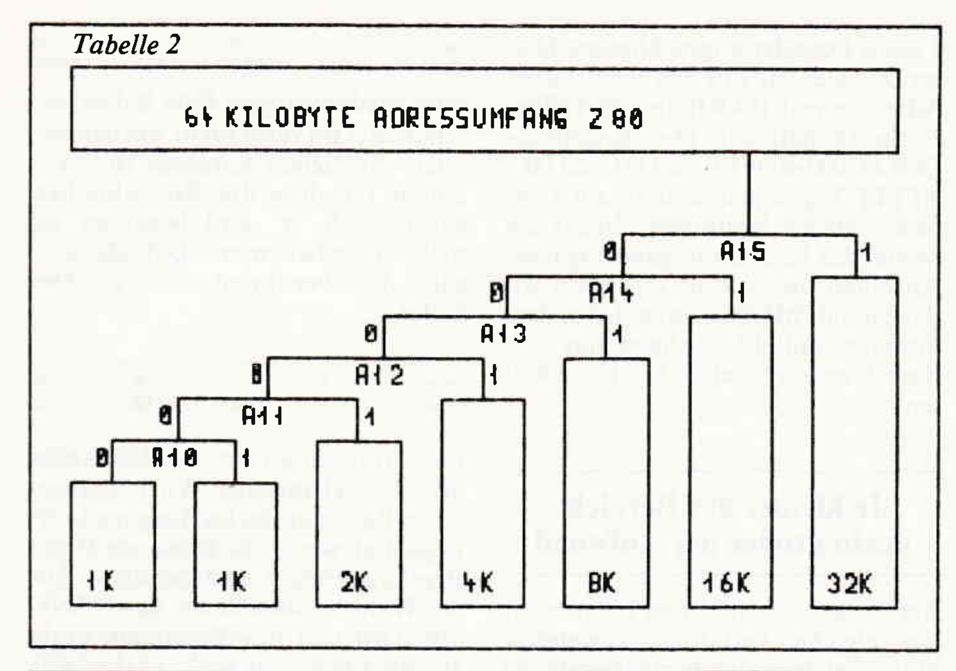


Bild 1

Tabelle 2



Adressen																	
		Adressbit's A0-A15															
DEZ	HEX	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0000													0	0	0	0
15	000F	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
16	0010													0	0	0	0
31	001F	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
32	0020												0	0	0	0	0
63	003F	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
64	0040											0	0	0	0	0	0
127	007F	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
128	0080										0	0	0	0	0	0	0
255	00FF	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
256	0100									0	0	0	0	0	0	0	0
511	01FF	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
512	0200								0	0	0	0	0	0	0	0	0
1023	03FF	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
1024	0400							0	0	0	0	0	0	0	0	0	0
2047	07FF	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2048	0800						0	0	0	0	0	0	0	0	0	0	0
4095	0FFF	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
4096	1000					0	0	0	0	0	0	0	0	0	0	0	0
8191	1FFF	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
8192	2000				0	0	0	0	0	0	0	0	0	0	0	0	0
16383	3FFF	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16384	4000			0	0	0	0	0	0	0	0	0	0	0	0	0	0
32767	7FFF	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32768	8000		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
65535	FFFF	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabelle 4

A15	A14	A13	Q7	Q6	Q5	Q4	Q3	Q2	Q1	Q0	AUSGAENGE
0	0	0	1	1	1	1	1	1	1	0	PIN 15
0	0	1	1	1	1	1	1	1	0	1	14
0	1	0	1	1	1	1	1	0	1	1	13
0	1	1	1	1	1	1	0	1	1	1	12
1	0	0	1	1	1	0	1	1	1	1	11
1	0	1	1	1	0	1	1	1	1	1	10
1	1	0	1	0	1	1	1	1	1	1	9
1	1	1	0	1	1	1	1	1	1	1	7

zwei Zahlensysteme anzutreffen. Durch die binäre Struktur des Systems (ein/aus; Spannung/keine Spannung) bietet sich das Binärsystem geradezu an. Weil man aber sehr viele Ziffern benötigt, um z.B. die dezimale Zahl 32768 darzustellen, verwendet man im allgemeinen das hexadezimale Zahlensystem, kurz HEX genannt. Hier werden die Ziffern größer 9 als die Buchstaben A – F gekennzeichnet. Die Basis dieses Zahlensystems ist die Zahl 16, im Gegensatz zur 10 im Dezimalsystem.

Wollen wir das in unser System umsetzen, so müssen wir wissen, daß jede unserer Adressleitungen eine Binärzahl repräsentiert. Wir fassen je vier dieser Leitungen zu Gruppen zusammen (4 Bit=1 Nibble; 4 Nibble = 16 Adressbits = Breite unseres Adressbusses).

Mit diesen vier Gruppen zu je vier Bit kann man z.B. die Zahlen HEX -8,0,0,0- binär als Bitmuster -1000,0000,0000, 0000- darstellen.

Wie Sie aus Tabelle 2 ersehen können, entspricht das Bit, das auf 1 gesetzt ist, genau dem, welches den Speicherbereich in zwei Hälften teilt. Da wir unseren "Musterspeicher" von 8-Kilobyte-Typen ausgehend decodieren wollen, müssen wir die Adressleitungen A15-13 in unsere Decodierung einbeziehen. Die restlichen Leitungen benötigt der Speicher, um die richtigen Zellen anzusprechen (2^ 13 = 8192 = 8 KB). Wir legen den Speicheranfang ab Adresse HEX (&)C000 mit 16 KB Größe fest. Die Spezis unter Ihnen haben sofort gemerkt, daß dies die Adresse des Bildspeichers ist. Zuerst rechnen wir &C000 in die dazugehörige Dualzahl um, um die Zustände der Adressleitungen herauszufinden.

HEX	C	0	0	0
DUAL	1100	0000	0000	0000
	A15-12	A11-8	A7-4	A3-0

Die Umwandlung HEX--> Dual legen Sie schon jetzt im Hinterstübchen ab, denn diese brauchen wir für unsere Arbeiten noch sehr oft.

UND ODER NICHT, das ist hier die Frage!

In der Digitaltechnik bedient man sich ganz besonderer Schaltungen. Sie werden benötigt, um bestimmte Probleme nach einem ganz bestimmten Muster zu lösen. Drei der wichtigsten Schaltfunktionen möchte ich Ihnen hier vorstellen: die NICHT-, die UND- und die ODER-Funktion. Mit diesen drei Grundlogikgliedern,

wie man sie nennt, kann man alle digitalen Schaltglieder bilden, die überhaupt nur denkbar sind. Logische Schaltglieder sind integrierte Bausteine, die in sogenannter TTL-Technik (Transistor-Transistor-Logik) aufgebaut sind. Solche Bausteine gestalten ihre Ausgangszustände (einen oder mehrere) in Abhängigkeit von ihren Eingangszuständen (einen oder mehrere) nach genau festgelegten Gesetzen. Solche Gesetzmäßigkeiten bringt man in sogenannten Wahrheitstabellen zum Ausdruck, oder man kann sie auch in mathematische Funktionen kleiden. Die Grundschaltungen der Logikbausteine entnehmen Sie bitte Bild 2.

Aus 16 mach 2

Rufen wir uns noch einmal die Adresse &C000 ins Gedächtnis. Hier waren die Bits A15,A14 auf 1 und A13 auf 0 gesetzt. Da wir nur 8 KB Speicher maximal adressieren, benötigen wir genau 13 Adressbits (A0-12, siehe Tabelle 2). Ebenso müssen wir dafür sorgen, daß 2 "CS"-Signale erzeugt werden, nämlich dann (und nur dann), wenn die Adressen &C000 - &DFFF und &E000 - &FFFF ausgegeben werden und das Prozessorsignal -MREQ- durch LOW einen Speicherzugriff anzeigt.
Die Halbleiterindustrie stellt uns für solche Aufgaben leistungsfähige Bausteine zur Verfügung. Einer dieser Wunderkäfer ist der "74 LS 138", ein 1 aus 8 Decoder. In Tabelle 4 sehen Sie die Wahrheitstabelle dieses Bausteins.
Bei diesem Decoderbaustein ist immer nur derjenige Ausgang auf low, dessen zugehörige Bitkombination am Eingang anliegt (siehe Wahrheitstabelle). Wie schon erwähnt, kann man mit drei Leitungen die Zahlen 0 – 7 darstellen. Diese Tatsache nutzt man bei dieser Art von Dekodierung aus. Die Adressleitungen werden an einen 1 aus 8 Decoder angeschlossen. Dieser Decoder teilt uns den verfügbaren Adressbereich (64 KB) in acht Teilbereiche (8 KB) auf. Die Anschlüsse Q6=&C000-&DFFF und Q7=&E000-&FFFF liegen genau dann auf Low, wenn wir sie benötigen. Zusätzlich besitzt das IC drei Freigabeeingänge. An einen dieser Pins schließen wir das Signal "MREQ" an, welches dem Decoder mitteilt, er habe das CS-Signal entsprechend durchzuschalten.

Je kleiner der Bereich, desto größer der Aufwand

Bei Ausgabeoperationen können wir das gleiche Verfahren anwenden. Nur wird hier der Freigabeeingang mit dem Signal "IOREQ" verbunden, um auf einen Zugriff hinzuweisen. Für diese Betriebsart besitzt der Prozessor Z80 spezielle Assemblerbefehle. Von Basic aus wird dieses Signal mit den Befehlen INP und OUT aktiviert.
Unsere letzten Betrachtungen bezogen sich auf Speicheroperationen mit Speichergrößen von 8 KB. Da man aber bei Ausgabeoperationen bei weitem keine 8 KB benötigt, müssen wir den angesprochen Bereich sehr verkleinern. Der Decoder 74138 hat einen Adressierbereich von sieben Adressen. Um genau sieben Adressen aus einem beliebigen Bereich anzusprechen, müssen wir dafür sorgen, daß der Decoder nur in diesem Bereich aktiv ist. Das Betriebssystem der Schneider Computer nutzt die Möglichkeit des Z80, den gesamten Adressbus dazu zu verwenden, eine Ausgabeadresse zu bestimmen. Unsere Baugruppen liegen mit den Ausgabeadressen im Bereich von &F9E0 – &F9EF. Interessant sind für uns erst einmal die sieben Adressen von &F9E0 - &F9E7. Zunächst rechnen wir die gewünschte Adresse in ihre Dualdarstellung um.

HEX	F	9	E	0
DUAL	1111	1001	1110	0000

Vom niederwertigen Byte haben wir schon drei Bit verbraucht, um den Bereich von sieben Adressen zu dekodieren. Für diese drei Bits schreiben wir einfach "x" und beachten sie nicht weiter bei unserer Dekodieraufgabe, das übernimmt schon der Dekoder.

HEX	F	9	E	0
DUAL	1111	1001	1110	0xxx

Jetzt erinnern wir uns des Abschnitts über Logikbausteine. Wir benötigen einen Baustein, der im Ausgang Low-Pegel hat, wenn alle Eingänge High-Pegel annehmen, und nur dann. Aus der Wahrheitstabelle für das NAND sehen wir, daß diese Bedingung exakt zutrifft. Ein geeigneter IC ist das 7430, ein 8-fach NAND (8 Eingänge). Nun schließen wir an jeden Eingang ein Adressbit mit dem Pegel 1 an. Das 7430 besitzt nur acht Eingänge, wir nehmen, großzügig wie wir sind, einfach zwei dieser billigen Käfer. Jetzt bleiben noch die Adressbits A4, A9 und A10 übrig, die wir mittels Inverter noch auf den geforderten Eingangspegel von 1 bringen müssen. Mit einem NAND-Baustein kann man, wenn alle Eingänge zusammengeschaltet sind (gleiche Pegel laut Tabelle), Inverter realisieren. Dazu nehmen wir einen weiteren Universal-IC, den 7400 (4 NAND mit je 2 Eingängen).
Die Ausgänge Q1 und Q2 haben, wenn die geforderten Adressen angesprochen werden, beide Low-Pegel. Wir müssen daraus ein Signal formen, das im Ausgang nur dann low ist, wenn auch die Eingänge low sind. Diese Aufgabe erfüllt ein ODER-Gatter (7432). Somit haben wir an Q3 ein geeignetes Signal, das den Decoder 74138 zum richtigen Zeitpunkt freigeben und sperren kann.
Nun haben wir schrittweise sieben Adressen aus einer Anzahl von 65536 herausgefischt. Nach Schaltbild 1

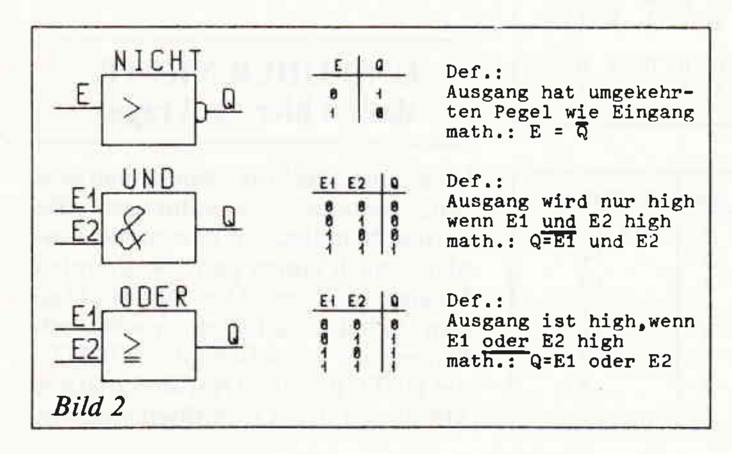
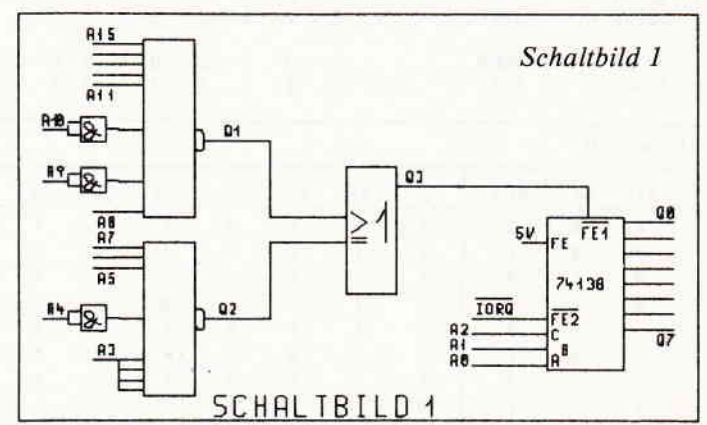


Bild 2



Schaltbild 1

können Sie eine eigene Adressdecodierung mit entsprechenden Freigabebedingungen aufbauen.
Die in diesem Beitrag gebotenen Grundlagen sind mit Sicherheit ein wichtiger Grundstock für alle diejenigen unter Ihnen, die sich mit Hardwareerweiterungen beschäftigen wollen.
Die erste Schaltung, die voraussichtlich im nächsten Heft vorgestellt werden wird, ist eine universell einsetzbare und programmierbare Parallelschnittstelle. (P. Richter/ME)

Speicherplatz = Speicherplatz?

Diese Betrachtung findet im Rahmen der Nachlese der Erfahrungen statt, die wir auf der CeBit im Gespräch mit unseren Lesern gemacht haben.

Was ist Speicherplatz?

Speicherplatz ist schlicht und einfach der Platz, der Informationen in Form von Bytes enthalten kann; demzufolge wird der bei einem Rechner zur Verfügung stehende Speicherplatz in Byte oder Kilo-Byte angegeben, wobei aufgrund der binären Arbeitsweise aller Rechner ein kByte nicht 1000, sondern 1024 Bytes enthält. Somit sind 64 kByte 64x1024 = 65536 Bytes (einzelne Speicherplätze).
Dieser Speicherplatz aber teilt sich auf in Read Only Memory (ROM); das ist der Speicherplatz, der nur ausgelesen werden kann, in dem also keine eigenen Programme untergebracht werden können, und in Random Access Memory (RAM), den Speicherplatz, auf den der Programmierer frei zugreifen kann.
Im ROM sind z.B. das Betriebssystem des Rechners und bei den CPC's die Sprache Basic gespeichert; diese bleiben auch nach dem Ausschalten des Rechners erhalten und können auch nicht durch Überschreiben gelöscht werden.
Der RAM-Bereich ist nun der Speicherplatz, der als "freier" Speicherplatz bezeichnet wird, auf den der Programmierer beliebig durch Schreiben und Lesen zugreifen kann, und in dem auch Basic- und Assemblerprogramme abgelegt werden. Die in diesem RAM abgelegten Informationen allerdings werden beim Ausschalten des Rechners gelöscht; die Speicherung dieser Daten muß also in Form magnetischer Aufzeichnung auf Kassette oder Diskette geschehen.
Dieser RAM-Bereich ist der eigentlich interessante Speicherplatz für Programmierer; er besteht im CPC aus ca. 42 kByte. Ein Programm, welches mehr als den zur Verfügung stehenden Speicherplatz benötigt, muß also theoretisch zerteilt und teileweise abgespeichert und wieder aufgerufen werden; speziell bei Kassettenbetrieb ist dieses Verfahren äußerst zeitaufwendig. Hierdurch entsteht recht schnell der Wunsch, einen größeren Raum zur Unterbringung eigener Programme zur Verfügung zu haben. Die offensichtliche Lösung des Problems heißt: Speichererweiterung.

Was ist Speichererweiterung?

Speichererweiterung ist schlicht und einfach eine dem Programmierer zusätzlich zum Speicher des Rechners zur Verfügung stehende Menge an Speicherplatz. Diese Erweiterungen stehen in verschiedenen Größen zur Verfügung; grundsätzlich sinnvoll sind Blöcke von 32 oder 64 kByte RAM. Eine solche RAM-Karte wird zumeist per Erweiterungsanschluß mit dem Rechner verbunden und erlaubt dem Prozessor beliebige Zugriffe auf ihren Inhalt.
Spätestens jetzt muß die Leistungsfähigkeit des Prozessors betrachtet werden, um nicht dem Irrglauben zu verfallen, man könne aus seinem 464 eine mächtige MegaByte-Maschine machen, die im Preis deutlich unter dem eines mit dem gleichen Speicherplatz ausgerüsteten Micro- oder Minicomputers liegt.

16 bit = 2 mal 8 bit?

Der in den CPC's enthaltene Prozessor Z-80 ist ein sogenannter 8-Bit-Prozessor. Diese Angabe sagt aus, daß dieser Prozessor acht Bit (ein Byte) breite Informationen auf einmal verarbeiten kann. Gleichzeitig hat der Prozessor 16 Adressenleitungen zur Verfügung, mit denen er jede einzelne Speicherstelle ansprechen kann. Und hier liegt auch schon die Beschränkung: bei 16 Adressenleitungen, die im binären System eine "0" oder eine "1" enthalten können, kommen wir mit der Formel: zwei (Basis des Zahlensystems) hoch 16 (Anzahl der Leitungen) auf die magische Zahl 65536 = 64 kByte.
Ein "8-Biter" kann also nur 65536 Speicherstellen ansprechen, während ein "16-Biter" mit 24 Adressleitungen (z.B. der Prozessor 68000) 2 hoch 24 = 16 MByte gezielt anspricht.
Dieses Problem macht sich schon beim CPC 6128 bemerkbar: Der Programmierer muß, will er die zweite "Bank" von 64 kB mitbenutzen, ständig zwischen der ersten und der zweiten Bank hin und her schalten. Hierzu muß eine spezielle Steuersoftware vorhanden sein, die dem Benutzer das Umschalten von einem Speicherbereich auf den anderen komfortabel ermöglicht. (Das Steuerprogramm für den 6128 heißt BANKMANager und ist auf der Systemdiskette enthalten.) Auch der Joyce, der in der Grundversion über 256 kB Speicherplatz verfügt, kann nur 64 kB auf einmal ansprechen. Die restlichen 192 kB (unterteilt in drei mal 64 kB) werden hier als "RAM-Floppy" benutzt; kann ein CPC-Besitzer die Laufwerke A und B anwählen, so steht dem Joyce noch das Laufwerk M (Memory) zur Verfügung, welches eben nur aus RAM besteht und seine gespeicherten Informationen nach dem Ausschalten des Rechners wieder vergißt.
Natürlich ist es möglich, die CPC's im Speicherplatz aufzurüsten; jedoch darf bei der Planung einer Erweiterung nicht außer acht gelassen werden, daß z.B. ein 80 kB langes Programm nicht einfach beim Überlauf des ersten Speichers in den zweiten geschrieben wird, sondern der Befehl, jetzt eine andere Bank zu benutzen, muß stets vom Programmierer kommen. Die Lösung dieses Problems kann nur lauten, das Programm in etwa 64 kB lange Module zu unterteilen, die auf verschiedene Bänken verteilt werden und sich mittels Steuersoftware selbst aufrufen.

Anwendungsmöglichkeiten

Wie eben besprochen, kann ein langes Programm von Kassette oder Diskette in die einzelnen Speicherbänke geladen werden und dort arbeiten. Ebenso können Daten "en masse" (z.B. die Adressen aus einer Adressenverwaltung) geladen und komfortabel bearbeitet werden. Der große Vorteil einer Speichererweiterung liegt in der hohen Zugriffsgeschwindigkeit, die die des Floppylaufwerks um ein Vielfaches übersteigt. Allerdings sind die im RAM gespeicherten Informationen beim Ausschalten unwiederbringlich verloren (es sei denn, die Erweiterung ist akkugepuffert).
Für kleine, zeitunkritische Anwendungen scheint deshalb die Arbeit mit dem Floppylaufwerk sinnvoller; eine beidseitig bespielte Diskette hat immerhin auch 340 kB Informationen gespeichert, Programmteile und Daten aller Art können ebenso komfortabel nachgeladen werden – man muß sich nur mit der Lese- und Schreibgeschwindigkeit des Floppylaufwerks arrangieren.
Die Entscheidung liegt bei Ihnen.
(ME)



Es gibt nur wenige, die es sich l
so viel zu zeigen. S
STIFTUNG WARENTEST
test 1/86
gut
Schneider CPC 464
TEXTOMAT
PROFIMAT
MATHEMAT
DATAMAT
BUDGET-MANAGER ZUM CPC
CBASIC Compiler
DR Draw
Schneider COMPUTER DIVISION
Schneider COMPUTER SOFTWARE
CUBIT
Golf
FLUGSIMULATOR
Schneider CPC 464
Benutzer-Handbuch
DDI-1
Benutzerhandbuch
CPC6128
Bedienungs-Handbuch
Der Schneider Computerkurs

Einführung in PASCAL
Teil 2

Nachdem im ersten Teil dieses Kurses eine kleine Exkursion in die Geschichte der Programmiersprachen unternommen worden war, soll sich nun dem eigentlichen Zweck, dem Programmieren in der Sprache Pascal und den dazu notwendigen Grundlagen, zugewandt werden. Dazu gehört auch, daß man sich über die Zielsetzungen beim Einsatz von Computern und ihrer Programmierung im klaren ist.

Wozu ein Programm?

Beweggründe zur Erstellung eines Computerprogrammes können die Automatisierung eines Vorgangs, die Lösung eines Problems oder, was sich anhand der unzähligen Computerspiele nicht verleugnen läßt, auch der Spaß an der Freude sein – was sich unter anderem auch als ein Problem darstellen kann. So unterschiedlich auch der Anlaß sein mag, die daraus resultierenden Programme haben alle etwas gemein: sie wurden so entworfen, daß ihre Abarbeitung durch den Computer ein genau bestimmtes Ergebnis zur Folge hat – zumindestens sollte es so sein.

Algorithmen

Wie auch immer, die Lösung der gestellten Aufgabe, egal welcher Art, wird mit einer Folge von Einzelschritten ausgeführt, auch Algorithmus genannt. Nach dieser mit einem so fremd klingenden Wort umschriebenen Vorgehensweise erledigen wir meistens unbewußt alle unsere Tätigkeiten – schrittchenweise.

Beispiel: Kaffee kochen
1. Wasser in Kaffeemaschine füllen.
2. Kaffeepulver in Filtertüte geben.
3. Kanne und Filter in/an Kaffeemaschine stellen.
4. Maschine einschalten.
5. warten...

Der erfahrene Kaffeetrinker wird sofort erkennen, daß obiger Algorithmus nur einer von mehreren möglichen ist. Z.B. ist mit Kaffeemaschinen gekochter Kaffee nicht jedermanns Sache. Ebenso dürfte der Algorithmus für jemanden, der mit den örtlichen Begebenheiten unvertraut ist, nicht ausführbar sein. Er müßte um einige Schritte erweitert bzw. verfeinert werden, die beispielsweise folgendes angeben:
1. Wo ist das Wasser zu holen und wieviel?
2. Die gleichen Angaben werden auch für das Kaffeepulver benötigt,
3. Stellt man die Kanne auf den Filter oder umgekehrt?

usw.

Anhand dieses Beispiels ist schon eine Methodik der Programmierung zu erkennen: einen Algorithmus erst grob zu skizzieren und dann immer weiter zu verfeinern, bis er schließlich ohne Wenn und Aber zum gewünschten Ziel führt. Dieses schrittweise Vorgehen – wie war das mit dem Algorithmus doch gleich? – wird "Top-Down" (das ist Fachtermologie und heißt soviel wie "von nun an geht's bergab") oder zu Deutsch, als "schrittweise Verfeinerung" bezeichnet.

Vorarbeiten

Bevor man sich also Zwecks der Programmierung an den Computer begibt und ihn mit Augenblicks-Einfällen füttert, sollte man sich über den einzuschlagenden Lösungsweg (Algorithmus) ohne Zweifel im klaren sein. Je mehr Energie man auf den Entwurf des Algorithmus verwendet, umso weniger Zeit benötigt man dann bei der eigentlichen Programmier-/Codierarbeit. Für diesen Zweck können verschiedene Hilfsmittel verwendet werden. Zum einen kann man einen Algorithmus verbal beschreiben – wozu im allgemeinen keine fundierten Programmierkenntnisse erforderlich sind (s. Kaffeekochen). Andererseits gibt es unterschiedliche grafische Darstellungsmöglichkeiten Algorithmen. Ein altes Hilfsmittel ist der Programmablaufplan, welcher langsam wegen fehlenden Strukturierungsmöglichkeiten und Unübersichtlichkeit aus der Mode kommt. "In" ist dagegen das "Nassi-Schneiderman-Diagramm" (Strukturdiagramm, Struktogramm). Wie die deutsche Bezeichnung schon vermuten läßt, ist dieses Hilfsmittel besonders gut zur grafischen Darstellung von Algorithmen geeignet.

Während dieser Zeit bleibt der Computer ausgeschaltet – es sei denn, er stellt in Form geeigneter Software Hilfsmittel zu diesem Zweck zur Verfügung.

Daten

Außer dem Algorithmus müssen auch die zu verarbeitenden Daten logisch zusammengestellt und gegliedert werden, z.B. in Datenstrukturen wie Tabellen, Listen oder Matrizen; oder die Zusammenfassung von Daten wie Name, Straße, Wohnort und Telefonnummer zu einer zusammenhängenden Datenstruktur namens Adresse.

Ist diese Phase erfolgreich abgeschlossen, so wird der Algorithmus von einem Programmierer mit Hilfe einer Programmiersprache in ein Programm übersetzt. Dabei kann die Übersetzungsarbeit auf ein Minimum reduziert werden, wenn der Algorithmus in entsprechender Form entworfen wurde und vorliegt.

Damit wären wir jetzt wieder beim Kern der Sache. Die Programmiersprache Pascal wurde nämlich genau zu diesem Zweck entwickelt: das Übertragen von Algorithmen zu vereinfachen, was besonders bei der Verwendung von Strukturdiagrammen zu erkennen ist, sowie den Aufbau von komplexen Datenstrukturen zu ermöglichen. Ersteres wird mittlerweile auch von manchen Basic-Implementationen unterstützt, Letzteres meines Wissens leider nicht.

Das erste Programm

Damit nun der Pascal-Compiler endlich etwas zu tun bekommt – wenn auch noch nicht sehr viel – hier das erste, einfache Programm (Listing 1). Dieses Programm wird bei seiner Ausführung zwei durch ein Leerzeichen getrennt eingegebene Zahlen einlesen und die Summe dieser Zahlen wieder ausgeben. Dem Basic-Programmierer wird wahrscheinlich sofort der Gedanke kommen, daß er das gleiche Resultat mit einer einzigen Programmzeile erzielen kann, im Gegensatz zu den neun Zeilen des Pascal-Programms. Dies liegt daran, daß wir das Programm besonders formatiert haben, um es übersichtlich zu halten. Ebenso hätte man das Programm auch in drei Zeilen unterbringen können (s. Listing 2), was aber sehr viel schwerer zu entziffern ist, wie man sieht.

Ebenso auffällig ist die unterschiedliche Groß- und Kleinschreibung der im Programm enthaltenen Wörter – in Pascal können Groß- und Kleinbuchstaben nach Belieben verwendet werden. Wie auch andere Program-

miersprachen, besteht Pascal aus einer Anzahl von reservierten Wörtern, auch Standardbezeichner genannt. Bei dem ersten Wort **Program** handelt es sich um ein solches. Das Wort **Summe** ist dagegen ein vom Programmierer definiertes Wort und klein geschrieben. Dies trägt zur leichteren Unterscheidung der reservierten Wörter bei.
An diesem Beispiel sollen nun einige der Grundeigenschaften der Sprache Pascal beleuchtet werden. Jedes Pascal-Programm beginnt mit der Programmdefinition, auch als Programmkopf bezeichnet. Hier die Zeile:
PROGRAM summe (input, output);
Dadurch wird dem Compiler mitgeteilt, daß die folgenden Zeilen ein Programm namens "summe" bilden. Ebenso wird dem Compiler in dieser Zeile mitgeteilt, daß das Programm auf externe Daten zugreifen soll, die es aus der Datei INPUT (Eingabe) erhält und Daten in die Datei OUTPUT (Ausgabe) schreibt. Diese, dem Programmnamen folgende Dateideklaration (INPUT, OUTPUT) sorgt also dafür, daß unser Programm mit der Außenwelt kommunizieren kann. Beide Dateien sind als Text-Dateien vordefiniert und Ein-/Ausgabegeräten zugeordnet, wobei bei den meisten heutigen Implementationen standardmäßig die Datei INPUT der Tastatur und die Datei OUTPUT dem Bildschirm zugeordnet ist. Deshalb kann bei manchen Pascal-Implementationen, wie z.B. Turbo-Pascal, die Dateideklaration entfallen.
Abgeschlossen wird die Zeile durch ein Semikolon. Dieses Symbol zeigt dem Compiler das Ende der ersten Anweisung unseres Programmes an und dient zur Trennung aufeinanderfolgender Befehle wie, in Basic der Doppelpunkt. Es muß jeder Anweisung folgen – bis auf ein paar Ausnahmen, die wir noch kennenlernen werden.

BILD 1. BEFEHLSFOLGE

ANWEISUNG 1
ANWEISUNG 2
⋮ ⋮
ANWEISUNG n

```
Pascal
BEGIN
      Anweisung 1;
      Anweisung 2;
      Anweisung :
      Anweisung n;
END
```

Als nächstes teilen wir dem Pascalcompiler mit, daß wir zur Bewältigung unseres Problems drei VARiable Größen benötigen: "zahl1", "zahl2" und "ergebnis". Im Gegensatz zu Basic, wo der Programmierer in sein Programm neue Variablen nach Gutdünken und sich ergebender Notwendigkeit mangels konkreter Planung einführen kann, muß bei Pascal jede Variable vor ihrer ersten Verwendung dem Compiler genau erklärt werden: ihr Name und vor allem ihr Typ. In unserem einfachen Beispiel handelt es sich bei den zu verarbeitenden Zahlen um ganze Zahlen, was dem Compiler durch das englische Wort INTEGER mitgeteilt wird. Stößt der Compiler beim Übersetzungsvorgang auf ein ihm unbekanntes Wort, also weder ein reserviertes noch ein vom Anwender vereinbartes Wort, so bricht er mit einer entsprechenden Fehlermeldung ab. Am Anfang wird dieser Zwang zur Disziplin gerade bei Basic-Freunden auf Unverständnis stoßen. Wem aber schon einmal bei einem Basic-Programm der Fehler unterlaufen ist, lediglich zwei Buchstaben eines Variablennamens zu vertauschen, was mir dank Schnellschreibens ab und zu mal passiert, und nicht gleich bemerkt wird, und sich über das sonderbare Verhalten seines Programmes zu wundern beginnt, der wird diese Einrichtung zu schätzen lernen. Wie gesagt, Pascal läßt sich auf eine weitere Bearbeitung des Programmes erst gar nicht ein, während Basic einfach für die "neue" Variable den Wert Null annimmt und fröhlich weitermacht.
Damit wären dem Compiler alle nötigen Informationen bezüglich der zu verarbeitenden Daten – welche, woher sie kommen und wohin sie gehen – gegeben. Was fehlt, ist nun die eigentliche Verarbeitungsvorschrift, der Programmkörper. Dieser besteht (meistens) aus mehreren Anweisungen, die den Ablauf des Programms bestimmen. In unserem Beispiel wird das Programm nur einmal durchlaufen, wobei folgende Schritte nacheinander ausgeführt werden müssen: das Einlesen der Daten von der Tastatur (INPUT), die Verarbeitung (Berechnung der Summe) und die Ausgabe (OUTPUT) des Ergebnisses. Damit haben wir auch schon die einfachste Programmstruktur kennengelernt: die "Befehlsfolge" (Bild 1).
Der Anfang des Programmkörpers wird dem Compiler durch das Wort BEGIN mitgeteilt. Aus allen darauf folgenden Texten versucht der Compiler einen vom Computer ausführbaren Code zu erzeugen, bis er auf das Wort END, gefolgt von einem Punkt stößt, welches ihm das Ende des Programms anzeigt.
Hier soll auch noch einmal kurz auf die besondere Formatierung des Programms hingewiesen werden: Die Zeilen zwischen BEGIN und END sind alle etwas eingerückt worden. Dadurch wird die Struktur dieses Programmteils/Anweisungsblocks auf einen Blick deutlich hervorgehoben. Wie man später bei umfangreicheren Programmen leicht erkennen wird, hilft diese der Algorithmusstruktur entsprechende Programmstrukturierung bei der Übersetzungsarbeit und Klammerung von logischen Anweisungsblöcken mit BEGIN und END, vor allem, wenn diese ineinander geschachtelt sind.
Nun zu den drei Anweisungen unseres Beispieles:

1. Das Einlesen der zu verarbeitenden Daten:
   READ (zahl1, zahl2);

Damit wird dem Computer der Auftrag gegeben, zwei Werte von der Tastatur einzulesen (READ) und diese in den Variablen zahl1 und zahl2 abzulegen. Die genaue Form dieser Eingabeanweisung ist eigentlich:
READ (dateivariable, zahl1, zahl2);
Dabei bezeichnet "dateivariable" die Datei, von der die Eingabe gelesen werden soll. Diese braucht man aber nicht anzugeben, wenn es sich um die Standard-Ein-/Ausgabedateien **input** bzw. **output** handelt, wie in unserem Fall. Die Anweisung
READ (input, zahl1, zahl2);
bewirkt also das gleiche.

2. Die Verarbeitung der Daten:
   ergebnis := zahl1 +zahl2;

Hier wird der Computer veranlaßt, einen numerischen Ausdruck zu berechnen, **zahl1 + zahl2**, und das Ergebnis der Variablen **ergebnis** zuzuweisen. Jede Zuweisung eines Wertes, sei es eine Konstante oder der Inhalt

BILD 2. WIEDERHOLUNG

ZÄHLER = STARTWERT ... ENDWERT	
	ANWEISUNG 1
	⋮ ⋮
	ANWEISUNG n

```
PASCAL
FOR zähler := startwert TO endwert DO
BEGIN
    Anweisung 1
        ⋮
    Anweisung n;
END
```

BILD 3. GRUNDAUFBAU PASCALPGM

PROGRAMM NAME
DATEI-
DEKLARATIONEN

PROGRAMM-
KOPF

KONSTANTEN
VARIABLEN
(TYPEN, FUNKTIONEN
PROZEDUREN)

DEKLARATIONS-
VEREINBARUNGS TEIL

ANWEISUNGEN

PROGRAMMKÖRPER

einer Variablen zu einer anderen Variablen, wird durch den Zuweisungsoperator := (Doppelpunkt und Gleichheitszeichen) veranlaßt.
Hier wird ein weiterer Unterschied zu Basic deutlich, wo eine Zuweisung durch das Gleichheitszeichen erfolgt und für Anfänger leicht zu Mißverständnissen führen kann:

a = a + 1

hat nun mal nichts mit mathematischer Gleichheit zu tun.

3. Die Ausgabe des Ergebnisses:
WRITELN ('Die Summe von...)
Hier wird nun das Ergebnis der Addition auf dem Bildschirm ausgegeben. Wichtig ist dabei, daß auszugebende Texte wie "Die Summe..." in **einfache** Anführungszeichen und die Angabe der auszugebenden Variablen außerhalb dieser Zeichen aufgeführt werden und durch Kommas getrennt sind. Warum das so sein muß, erfahren wir später. Nach der Ausgabe wird dann auf dem Ausgabegerät eine neue Zeile begonnen (WRITE LiNe). Da wir keine Ausgabedatei in der Anweisung angegeben haben, wird, ähnlich wie bei der READ-Anweisung, als Ausgabedatei die Standard-Datei OUTPUT (=Bildschirm) vom Programm gewählt.
Anhand dieses kleinen Beispielprogramms wird schon die Schematik deutlich, nach der ein jedes Pascalprogramm aufgebaut/gegliedert ist (Bild 2). Dazu noch ein einfaches Programm mit (wenig?) praktischem Nutzen (Listing 3):
Dieses Programm liest fünf (maxwerte) Nettopreise ein, getrennt durch Eingabe der RETURN-/ENTER-Taste, ermittelt den Bruttobetrag und summiert diese auf. Diese Summe wird dann, da es nicht im Sinn der Sache ist, wenn der Computer Daten für sich behält, ausgegeben.
Als erster Unterschied fällt der in der zweiten Zeile stehende, mit (* und *) eingeschlossene Text auf. Hierbei handelt es sich um einen Kommentar, der zu näheren Beschreibungen/Erläuterungen von Programmen bzw. Teilen an jeder beliebigen Stelle eines Pascalprogramms eingefügt werden kann. Trifft der Compiler beim Übersetzen des Programmes auf einen solchen Text, so wird er einfach überlesen und beansprucht im erzeugten, lauffähigen Programm keinerlei Platz. Kenner von Basic kennen die Knausrigkeit im Einsatz von Kommentaren in ihren Programmen zu Genüge. Verbraucht doch jeder Buchstabe in Basic ein wertvolles Byte!
Des weiteren ist zu der VARiablendeklaration die Konstantenvereinbarung

CONST mwst = 1.14;
maxwert = 5;

hinzugekommen. Die Deklaration von in einem Programm benötigten Konstanten zu Beginn desselbigen birgt zwei wesentliche Vorteile:

1. Durch eine entsprechende Bezeichnung der Konstanten ist ihr Zweck im Programm auch nach einiger Zeit noch ohne detektivische Nachforschungen erkennbar. So gibt der Ausdruck **wert * mwst** weniger Rätsel auf als **wert * 1.14**, zumal bei der Konstantenvereinbarung auch noch ein kleiner Hinweis in Form eines Kommentares gegeben werden kann.
2. Es kommt oft vor, daß ein und derselbe konstante Wert an verschiedenen Programmstellen benötigt wird. Muß dieser Wert aus irgendeinem Grund einmal geändert werden, was z.B. bei der Mehrwertsteuer nicht auszuschließen ist, so braucht die Änderung nur an einer Stelle des Programmes vorgenommen werden. Ansonsten müßte das ganze Programm nach den betroffenen Stellen durchforstet werden, wobei die eine oder andere nötige Änderung übersehen werden kann.

Im Anweisungsteil selbst wurde als Beispiel eine weitere Programmstruktur angewandt: eine Form der Wiederholung, die Basic-Programmierern als FOR-NEXT-Schleife bekannt ist, in Pascal aber eine etwas andere Form hat:

FOR count=1 TO maxwerte DO

Durch diese Anweisung wird der Computer dazu veranlaßt, die nach

Listing 1:

```
PROGRAM summe (input, output);

VAR zahl1, zahl2, ergebnis : INTEGER;

BEGIN
   READ (zahl1, zahl2);
   ergebnis := zahl1 + zahl2;
   WRITELN ('Die Summe von ',zahl1,' und ',zahl2,' ist ',ergebnis);
END.
```

Listing 2

```
PROGRAM summe (input, ouput); VAR zahl1, zahl2, ergebnis : INTEGER; BEGIN READ
(zahl1, zahl2); ergebnis := zahl1 + zahl2; WRITELN ('Die Summe von ',zahl1,' und
',zahl2,' ist ',ergebnis); END.
```

Listing 3

```
PROGRAM brutto (input, output);
(* Erm. des Bruttopreises von maxwerte Nettopreisen *)
CONST   mwst = 1.14;
     maxwerte = 5;
VAR  count : INTEGER;
      wert,
     summe : REAL;
BEGIN
   summe := 0;
   FOR count := 1 to maxwerte DO
   BEGIN
      WRITE ('Wert ',count,' ? ');
      READLN (wert);
      summe := summe + wert * mwst;
   END;
   WRITELN ('Bruttopreis von ',maxwerte,' Artikeln ist ',summe);
END.
```

dem DO zwischen BEGIN und END liegende Befehlsfolge fünfmal (maxwerte) auszuführen. (Mehr zu einfachen und strukturierten Anweisungen gibt es im nächsten Teil.)

Zum Schluß noch einmal eine Zusammenfassung der bisher kennengelernten Eigenschaften eines Pascalprogramms:

- Jedes Programm beginnt mit dem Programmkopf, in dem ihm ein Name gegeben und die Art der Kommunikation mit der "Außenwelt", d.h. von wo es Daten bekommt und wohin es die erarbeiteten wieder schickt, festgelegt wird. Letzteres kann bei einigen Pascal-Implementationen entfallen.
- Im Deklarationsteil werden alle im Programm benötigten Größen wie Variablen, Konstanten und (darüber auch etwas später mehr) Typen, Funktionen und Prozeduren vereinbart.
- Der Programmkörper beinhaltet schließlich die zur Ausführung des Programmes nötigen Anweisungen.

Fortsetzung folgt... (MC)

Miniport am CPC

Mit der hier vorgestellten Schaltung ist es möglich, allerlei elektrisches Gerät über die Tapebuchse des 664/6128 ein- und auszuschalten. Am 464 ist dies nur nach Öffnen des Geräts und "Anzapfen" der Zuleitungen des Rekorders möglich - allerdings unter Garantieverlust.

Das Ganze besteht lediglich aus einer aus einem Transistor bestehenden Treiberstufe, die das vom Schaltausgang der Tapebuchse ausgehende Signal (ein/aus) zur Ansteuerung verschiedenster Verbraucher verstärkt.

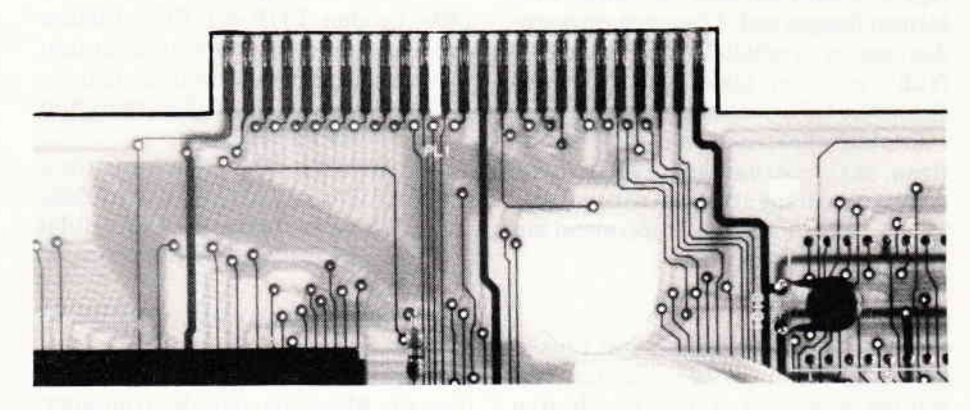
Hierbei sind allerdings einige Dinge zu beachten: Der der Buchse entnommene Strom sollte ca. 200 mA nicht überschreiten, ab ca. 100 mA Stromaufnahme des Verbrauchers (Last) sollte anstatt des Transistors BC 109 einer des Typs 2N 1613 Verwendung finden. Als Last können eine Lampe, ein kleiner 5-V-Motor oder ein Relais zur Ansteuerung von Netzverbrauchern (Kaffeemaschine!) angeschlossen werden.

Zusätzlich ist natürlich stets die Polarität der angeschlossenen Verbraucher zu beachten (siehe Skizze).
Wenn die fertige Schaltung angeschlossen ist, können externe Geräte mit 'call &BC6E' eingeschaltet und mit 'call &BC71' wieder ausgeschaltet werden.

Stückliste:
R1 = 330 OHM 0,25W
T1 = BC 109 oder
2N 1613
D1 = 1N 4001

(W. Wagner/ME)

Reset per Knopfdruck



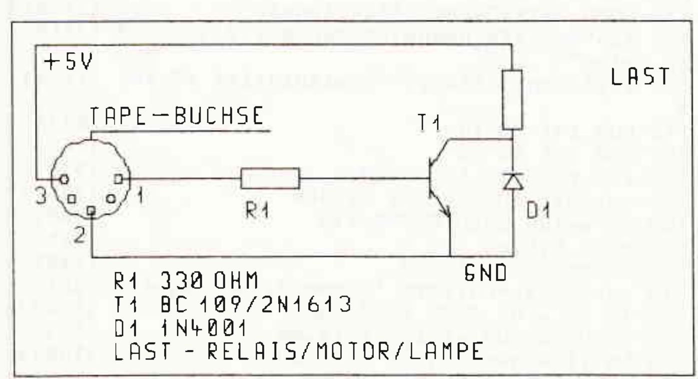
Dieser kleine Hardwaretip ermöglicht durch einen kleinen Eingriff in die CPC-Rechner das komfortable Kaltstarten der Maschine, wobei auch die meisten Spiele und Anwendungen, die den "Drei-Finger-Griff" (Ctrl-Shift-Esc) sperren, zurückgesetzt werden können.
Dieser Eingriff ist denkbar einfach: Die auf den Expansionport herausgeführten Leitungen "GND" (Pin 49) und "RESET" (Pin 41) werden mit je einem ca. 10 cm langen, isolierten Draht verlötet. An die beiden freien Drahtenden wird ein Taster "1x ein" (kein Schalter !) angelötet - fertig (siehe Skizze). Bei Betätigung des Tasters wird nun über einen 150 Ohm-Widerstand und einen Inverter die Resetleitung des Prozessors auf low-Potential gelegt und so direkt am Prozessor die Neuinitialisierung des Rechners veranlaßt.
Zum Taster ist zu sagen, daß dieser von guter Qualität sein sollte, um nicht durch übermäßiges Prellen den Reset-Eingang unnötig zu belasten.

Der Taster kann auch mit einer Bohrung auf der Oberseite der Rechner befestigt werden. Bei dem ganzen Eingriff geht natürlich auch die von Schneider gewährte Garantie verloren.

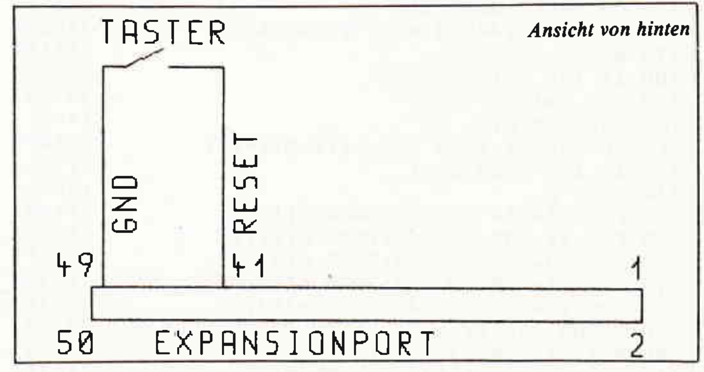
Bauteile:
Taster 1x ein
2 Drähte, ca 10 cm isol.

Werkzeug:
Lötkolben max. 30 W
Elektroniklot 1 mm 0

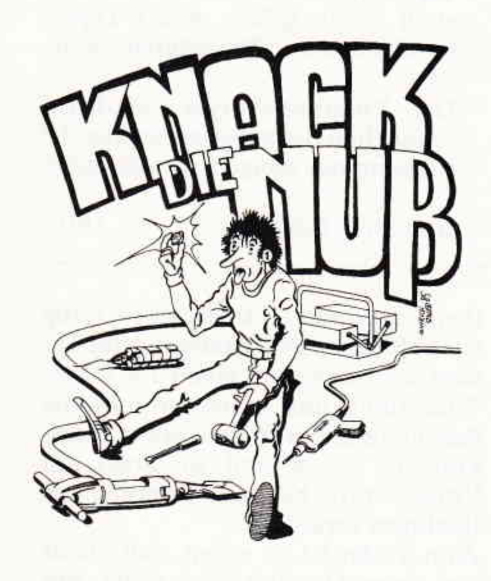
(S. Jügel/ME)



Skizze: Miniport



Skizze: Der Anschluß des Reset-Tasters



Diese Rubrik beschäftigt sich vornehmlich mit Problemlösungen in Basic. Die rege Resonanz auf unseren Aufruf, Problemstellungen und -Lösungen einzusenden, macht es möglich, daß "Knack die Nuß" zu einer ständigen Rubrik in Schneider CPC International wird. Auch weiterhin möchten wir alle Leser ermutigen, aktiv mitzuarbeiten. Die folgenden Kurzbeiträge stammen von unserem Leser, Herrn Friedrich Oppermann aus Peine. Er schreibt uns:

Das Problem

Gegeben sind die Zahlen von 1 bis 9, die ohne Wiederholung in den Variablen a, b, c, d, e und f enthalten sind.

Gesucht wird die Quersumme von 111 dieser Zahlenkette. Es werden lediglich die Operationszeichen der vier Grundrechenarten einschließlich der Potenzierung ausgetauscht. Dies wird durch umfangreiches POKEn realisiert.

Zum Programm:
Zeilen

- 120–130 Zuweisung der Variablen.
- 150 Zum Abfangen von ERR=6 (Overflow) ist eine Error-Routine notwendig, die in Zeile 130 jeweils ein Ohm-Zeichen ausdruckt.
- 160 p1=Endadresse vom Listing im Speicher
- 170 In der Ursprungsgleichung für x muß unbedingt das" "-Zeichen stehen. Es nimmt das Token 249 ein.
- 180 i=6 wird als Flag benutzt, sonst erfolgt beim ersten Durchlauf "unexpected Return".
- 190–230 Auslesen der Speicheradressen und Zuweisung in die indizierte Variable a(i) (DIM nicht notwendig!)
- 220 Iteration von i bewirkt bei i=6 eine Gleichsetzung von p zu p1 und somit Abbruch der Schleife.
- 240–280 In den FOR-NEXT-Schleifen werden die entsprechenden mathem. Operationszeichen gePOKEd, deren Token zwischen 244 und 248 liegen
- 290 Checkt, ob die Gleichung für x erfüllt wird. Ist x=111, wird Zeile 170 gelistet. Nach LIST erfolgt unweigerlich der Abbruch, also END.

Im direkten Modus kann nochmals auf Richtigkeit von x überprüft werden. Dieses kleine Exempel gibt Aufschluß über die Möglichkeiten der computerspezifischen Eigenschaften von PEEK und POKE, die bei umsichtiger Anwendung nicht sogleich zum Programmabsturz führen müssen.
Mit diesem aufgezeigten Algorithmus können viele Zahlenbeispiele versteckt und aufgedeckt werden. Ein Ausflug in die Kryptographie steht in Aussicht!

Das Problem

Eine der ältesten Problemstellungen beinhaltet das nun folgende Beispiel "Jägermeister". Vor vielen hundert Jahren rief ein König seinen Hofjäger zu sich und erteilte ihm folgenden Auftrag: Erlege für mich genau hundert Stück Wild. Es sollen Rehe, Hasen und Tauben sein.

Pro Reh gibt es 10 Taler.
Pro Hase gibt es 3 Taler.
Pro Taube gibt es 1/2 Taler.
Merke! Die Summe der hundert Tiere soll ebenfalls genau 100 Taler betragen.
Nun gibt es dafür nur eine richtige Lösung.

Zeile

- 100 REM
- 110 f$=flag
- 120 a$=Auslegung für PRINT USING
- 130 Preisstaffelung
- 140–160 Iterations-Schleifen zur Bestimmung der Wildmenge. Schätzungsweise können es nicht mehr als 10 Rehe, 20 Hasen und zwischen 50 bis 100 Tauben sein. Da nur volle Talerbeträge gelten, erfolgt STEP 2.
- 170–180 Bei w=100 wird die Subroutine aufgenommen.
- 200 Ausstieg, Lautzeichen und Ende, da Lösung parat.
- 210 Sa=Summe der einzelnen Beträge
- 220 Flag wird neu gesetzt.
- 230 Zwischenzeitlicher Ausdruck.
- 240 Das Statement IF wird vermieden durch einen logischen Vergleich. f$="Genau" ergibt -1. Multipliziert mit -1 = +1. Somit bietet sich die ON..GOTO..-Routine an.
- 250 Bei Nichterfüllung Rückkehr in die Schleife. F. Oppermann

Listing 1

```
100 REM Quersummen von 1 bis 9                [1288]
110 MODE 2:i=1                                [845]
120 DATA 1,23,45,67,8,9                       [889]
130 READ a,b,c,d,e,f                          [1104]
140 PRINT"Computer busy -> X=111"             [3518]
145 PRINT a;b;c;d;e;f                         [1058]
150 ON ERROR GOTO 310                         [1551]
160 p1=PEEK (&AE83)+256*PEEK(&AE84)           [1825]
170 x=a+b-c*d+e^f                             [1118]
180 IF i=6 THEN RETURN                        [1118]
190 FOR p=&170 TO p1                          [1285]
200 pp=PEEK(p)                                [412]
210 IF pp=249 THEN LET a(i)=p:i=i+1           [2309]
220 IF i=6 THEN p=p1                          [1123]
230 NEXT p                                    [364]
240 FOR z1=244 TO 248:POKE a(1),z1            [1106]
250 FOR z2=244 TO 248:POKE a(2),z2            [1359]
260 FOR z3=244 TO 248:POKE a(3),z3            [1881]
270 FOR z4=244 TO 248:POKE a(4),z4            [1577]
280 FOR z5=244 TO 248:POKE a(5),z5            [2671]
290 GOSUB 170:IF x=111 THEN PRINT:LIST 170    [2485]
300 NEXT z5,z4,z3,z2,z1                       [995]
310 PRINT CHR$(191);:RESUME NEXT              [1971]
```

Listing 2

```
100 REM Jaegermeister berechnet die Wildme [2402]
nge
110 MODE 2:f$="aber" 'f$=flag-$               [2456]
120 A$="Rehe:## Hasen:## Tauben:## \   \ #    [2419]
## Taler"
130 rr=10:hh=3:tt=0.5 'Preisstaffel (Taler    [1774]
)
140 FOR r=1 TO 10                             [647]
150 FOR h=1 TO 20                             [862]
160 FOR t=50 TO 100 STEP 2                    [974]
170 w=r+h+t 'Menge des Wildes                 [2081]
180 IF w=100 THEN GOSUB 210                   [888]
190 NEXT t,h,r                                [1018]
200 PRINT CHR$(7):END                         [1441]
210 sa=r*rr+h*hh+t*tt 'Summe d. Betrages      [2215]
220 IF sa =100 THEN f$="genau"                [1067]
230 PRINT USING a$;r,h,t,f$,sa                [1592]
240 ON (f$="genau")*-1 GOTO 200               [1804]
250 RETURN                                    [555]
```

Software von Lesern – für Leser

DIE GOLDENE 7

Secret of Wizard

Ein spannendes Abenteuer mit toller Handlung! Sie leben in Wizard, einem Land, in dem Magie noch Macht bedeutet. Ihr friedliches Dasein wurde abrupt unterbrochen, denn Angmar, der böse Hexenmeister aus dem Lande Forod Waith, treibt sein Unwesen. Er stürzte den Herrscher von Wizard und benutzt die Hauptstadt als Ausgangspunkt für seine Plünderungen. Die unterdrückte Bevölkerung hat Sie dazu auserwählt, den Diktator zu stürzen und Wizard den langersehnten Frieden zu bringen!

Turlen

Strategie- und Actionspiel mit mehreren Leveln! Turlen ist eine alte Stadt, die von feindlichen Roboterpanzern angegriffen wird. Schaffen Sie das Unmögliche, nämlich Turlen vor den feindlichen Angreifern zu schützen? Sie spielen entweder gegen den Computer oder gegen einen anderen Mitspieler! Turlen besitzt eine ausgezeichnete Grafik, guten Sound und hat mehrere verschiedene Bilder.
Mit Demomode!

Zodora

Sie übernehmen die Rolle eines Steuereintreibers, der zur Jahrhundertwende in einem verfallenen Schloß die letzten Goldstücke einsammelt, die auf einen früheren Reichtum schließen lassen. Die Jagd nach dem Schatz führt Sie über gefährliche Fallen und Förderbänder durch insgesamt 60 (!) verschiedene Bilder!

3-D Labyrinth

In den unterirdischen Gewölben von Istaki wimmelt es nur so von Geheimgängen. Überreste menschlichen Daseins zeugen von einem geheimnisumwobenen, labyrinthähnlichen Gebilde tief unter der Erdoberfläche. Vor Ihnen hat keiner dieses Labyrinth je wieder lebend verlassen. Es ist an Ihnen, das Geheimnis des Todeslabyrinthes zu lüften!



Die alte Burg

Hervorragendes Grafik-Adventure mit spannender Handlung. Ihr Onkel vererbte Ihnen ein altes Schloß in England. Hier soll, einer alten Sage zufolge, ein sagenhafter Schatz verborgen sein. Ihre Aufgabe ist es nun, diesen Schatz ausfindig zu machen. Doch Vorsicht – zahlreiche Gefahren begleiten Sie auf dieser Excursion.



Space-Race

Wer gewinnt das Space-Race? Mit Ihrem Raumschiff fliegen Sie durch insgesamt 10 Welten des Universums. Neben guten Nerven und einer schnellen Reaktion benötigen Sie vor allem viel Glück, um das Space-Rennen zu gewinnen. Denn Ihre Gegner versuchen mit allen Mitteln, Sie an Ihrem Unterfangen zu hindern! Schnelles Arcade-Spiel mit guter Grafik und Sound.

Galaxis

Auf dem Planeten Galaxis ist eine Seuche ausgebrochen. Sie sind der einzige überlebende Pilot und haben die ehrenvolle Aufgabe, die verschollenen Raumschiffe zu finden, um die verbliebenen Bewohner zu retten. Diese Aufgabe erfordert strategisches Vorgehen – zumal Ihr Raumschiff nur noch 2000 Liter Kraftstoff für diese Mission besitzt. Werden Sie der Retter des Universums?



Die Bezeichnung »Goldene Sieben« steht künftig für gute Software, die von den Lesern unserer Zeitschrift erstellt worden ist, und nicht als Listing im Heft abgedruckt werden kann, da hierfür nur einbegrenzter Raum vorhanden ist, und Ihre "Schneider CPC International" kein reines Listingblättchen sein soll!
Jeweils sieben tolle Programme sind auf der Kassette bzw. Diskette für wenig Geld enthalten.

Preise:

Kassettenversion	20,– DM
3"-Diskettenversion	30,– DM

zuzüglich 3,– DM Porto/Verpackung
(Im Ausland 5,– DM)
Bei Nachnahme kommt noch die die Nachnahmegebühr hinzu.
Ihre Bestellung richten Sie bitte mit dem nebenstehenden Coupon an:

**DMV Verlag
– Leserservice –
Postfach 250**

3440 Eschwege

7 Programme für CPC 464/664/6128

Bitte Bestellkarte benutzen

Die Gratis-Tools

Teil 5

Dynamisches Entwanzen mit DDT:
Ein wesentliches Element professioneller Softwareentwicklung ist der Test neu erstellter Programme. Im Unterschied zur Softwareentwicklung in BASIC nimmt kein Interpreter dem Programmierer eine Plausibilitätsprüfung der Programmzeilen ab. Auf Assembler-Ebene ist der Softwareentwickler der absolute Master über die CPU seines Systems. Er kann daher auch Programmsequenzen schreiben, die den Computer vollkommen "in die Wüste" schicken. In diesem Fall hilft oft nur noch das Ausschalten, um das System wieder betriebsbereit zu machen. Wenn eine in Maschinencode geschriebene Routine nicht das tut, was der Programmierer erwartet, dann ist es unbedingt notwendig, die Stelle im Programm zu finden, in der ein logischer Programmierfehler vorliegt.
Diese Möglichkeit erhält man durch das Werkzeug DDT (Dynamic Debugging Tool). Ins Deutsche übersetzt heißt DDT "Werkzeug zum dynamischen Entwanzen". Der angloamerikanische Begriff "Entwanzen" bedeutet im übertragenen Sinne "fehlerfrei machen". Dazu ist der Entwickler mit DDT in der Lage, indem er:
- das zu testende Programm in den Speicher laden
- den Speicherinhalt ansehen
- den Speicherinhalt verändern
- und das Verhalten des Programms auf unterschiedliche Weise untersuchen kann.

Das Arbeiten mit DDT soll anhand unseres Beispielprogramms EXA1. COM beschrieben werden.
Aufrufen von DDT:
Das Werkzeug läßt sich auf zweierlei Arten starten. Erstens: Indem der Name des zu ladenden und zu testenden Programms mit angegeben wird. Der Aufruf lautet dann:
DDT EXA1.COM (Return)
Zweitens: Indem man DDT ohne Angabe eines Programmnamens startet. Dann lautet der Aufruf:
DDT (Return)
DDT meldet sich nach dem Start wie folgt:
DDT VERS 2.2
Ist ein Programmname mit angegeben worden, dann prüft DDT, ob er dieses Programm in den Speicher laden kann. Gelingt dieses nicht, reagiert DDT mit einem Fragezeichen und seinem Prompt, einem Trennstrich. Also:
?
-
Immer wenn ein Prompt erscheint (-), ist DDT zum Befehlsempfang bereit. Die Befehle des DDT sind als Überblick im folgenden beschrieben. Wenn DDT das Programm laden konnte, sieht die Meldung folgendermaßen aus:
DDT VERS 2.2
NEXT PC
0180 0100
Die Einträge haben folgende Bedeutung:
NEXT bezeichnet, wieviele Bytes des Speichers durch das Laden des zu testenden Programms modifiziert wurden. In diesem Fall sind es die Bytes von 100 bis 17F. 180 ist das erste unveränderte Byte im RAM des Systems. Unser Programm EXA1.COM benötigt zwar nur acht Bytes, aber 128 Bytes sind das Minimum, das von CP/M bearbeitet werden kann. Deswegen wird ein Block von 128 Bytes geladen, wenn man EXA1.COM aufruft.
PC bezeichnet die Position des Program-Counters der CPU. In diesem Fall steht er auf der Startposition 100H.
Die DDT-Befehle:
1. File-Handling
I < name > "INPUT":
Eine Programmdatei zum Einlesen initialisieren.
< name > =Dateiname
R < offset > "READ":
Die mit "I" gekennzeichnete Datei einlesen.
< offset > =Versatz zwischen Originaladresse des Programms und gewünschter Startadresse in DDT.

2. Ablauf-Test
G < start > , < stop1 > , < stop2 > "GO":
Programmablauf starten.
< start > =Startadresse der Programm-Abarbeitung (default=PC).
< stop1 > =Stopadresse, bei der der Programmablauf anhalten soll, wenn sie erreicht wird.
< stop2 > =alternative Stopadresse.
T < times > "TRACE":
Programm in Einzelschritten abarbeiten und dabei den aktuellen Prozessorzustand anzeigen.
< times > =Anzahl der abzuarbeitenden Schritte (default=1).
U < times > "UNTRACE":
Programm in Einzelschritten abarbeiten, ohne den aktuellen Prozessorzustand anzuzeigen.
< times > =Anzahl der abzuarbeitenden Schritte (default=1).
X "EXAMINE":
Den Prozessorzustand anzeigen lassen.
X < register > "EXAMINE":
Den Zustand eines Registers anzeigen und gegebenenfalls ändern.
< register > =zu modifizierendes CPU-Register.
3. Speicher-Modifikation
D < start > , < stop > "DUMP":
Speicherinhalt hexadezimal und in ASCII auflisten.
< start > =Anfangsadresse des zu listenden Speicherbereichs.
< stop > =Endadresse des zu listenden Speicherbereichs.
L < start > , < stop > "LIST":
Speicherinhalt in Assemblercode auflisten.
A < address > "ASSEMBLE":
In Assemblercode geschriebene Daten in den Speicher schreiben.
S < address > "SET":
Den Speicherinhalt byteweise, hexadezimal anzeigen und gegebenenfalls ändern.
F < start > , < stop > , < value > "FILL":
Speicherbereich mit einem Datenwert füllen.
< value > =Füllzeichen (1 Byte).
M < start > , < stop > , < destination > "MOVE":
Speicherbereich verschieben.
< destination > =Anfangsadresse der neuen Lage des Datenblocks.
4. Rechenbefehle
H < value1 > < operator > < value2 >
"HEXADECIMAL COMPUTATION":
Einen hexadezimalen Wert berechnen.
< value1 > , < value2 > =maximal vierstellige, hexadezimale Werte
< operator > =Rechenbefehl, + für Summe, - für Differenz.
Einige DDT-Kommandos im Beispiel:
Im nächsten Abschnitt soll das Arbeiten mit DDT am Beispiel unseres ersten Übungsprogramms EXA1.COM näher beschrieben werden. Zunächst einige Hinweise zum Laden des zu testenden Programms. Sollte sich DDT mit dem Fragezeichen gemeldet haben, obwohl Sie den Programmnamen angegeben hatten, so gibt es folgende mögliche Ursachen:
Beim Angeben des Dateinamens beim Aufruf von DDT:
- die Extension.COM vergessen
- den Programmnamen falsch geschrieben
- das Programm nicht auf dieser Diskette gespeichert

Beim nachträglichen Laden:
- das Kommando I < name > vergessen oder falsch geschrieben (zwischen I und < name > darf kein Leerzeichen stehen, Extension muß mit angegeben werden).

Testen des Programmes mit Einzelschrittbearbeitung:
Geben Sie den Trace-Befehl ein:
T (oder t) (Return)
DDT sollte darauf mit folgender Zeile reagieren (Bild 1):
Was bedeutet das? C0Z0M0E0I0 ist kein Losungswort aus einem Adventure-Game, sondern stellt den Inhalt des Prozessor-Flagregisters dar. Der Buchstabe bezeichnet dabei das jeweilige Flag, die darauffolgende Ziffer die Stellung des Flags 0=reset, 1=set. Die Flags bedeuten im einzelnen.

C - CARRY
Übertrags-Flag, zeigt Übertrag im Akku an.
Z - ZERO
Null-Flag, zeigt Null im Akku an.
M - MINUS
Vorzeichen-Flag zeigt negative Zahlen im Akku an.
E - EVEN PARITY
Paritäts-Flag zeigt gerade Zahl im Akku an.
I - INTERDIGIT CARRY
Zeigt Übertrag vom Lower- zum Higher-Nibble im Akku an.

Die weiteren Einträge:
A - Akku
B - Registerpaar BC
D - Registerpaar DE
H - Registerpaar HL
S - Stack-Pointer
P - Program-Counter

Beim Trace des ersten Prozessorbefehls stellt man fest, daß alle Flags und Register den Wert Null enthalten, abgesehen vom Program-Counter, der auf die Speicheradresse 100 zeigt. Auch der Stack-Pointer zeigt auf 100. Da der Stackpointer dekrementiert wird, bevor ein neuer Wert auf den Stack geschrieben wird, werden Einträge in den Stack ab Adresse FF vorgenommen. Rechts neben der Angabe des P-Registers steht der disassemblierte Befehl, in diesem Fall:
MVI C,02
Auf diesen Befehl folgt ein Asterisk und danach der neue Programmzählerstand, also:
***0102**
Daraus ist zu entnehmen, daß der nächste auszuführende Befehl unter der Speicheradresse 102 steht.
Führen Sie den nächsten Trace-Befehl aus, das Ergebnis sollte sein (Bild 2):
Wir finden, daß der vorige Befehl den Wert 02 in das Register C eingetragen hat, wie die Trace-Zeile unter B beweist (B=0002). Der nächste Trace zeigt, daß das Kommando CALL 0005 ausgeführt wurde. Die Folge ist, daß als nächster Programmzählerstand die Adresse 0005 eingetragen wird, also (Bild 3):
Gleichzeitig sieht man, daß das E-Register den Wert 51 (für ASCII "Q") angenommen hat. Der nächste Trace führt zu folgender Anzeige (Bild 4):
An dieser Stelle haben wir uns in das BDOS von CP/M begeben, es werden also Befehle angezeigt, die nicht Bestandteil unseres kleinen Beispielprogramms sind. Der Trace zeigt uns, daß es auf Adresse 8F00 weitergehen soll. Gleichzeitig beobachten wir, daß der Stack-Pointer einen Eintrag über zwei Bytes vorgenommen hat, denn er zeigt nun auf 00FE. Die Ursache hierfür ist der vorangegangene CALL-Befehl. Ein Unterprogrammaufruf führt immer dazu, daß sich die CPU die Rückkehradresse auf dem Stack merkt. Wir wollen sehen, was sich der Prozessor dort eingetragen hat. Dazu benutzen wir das Kommando "DUMP" des DDT. Bitte geben Sie ein:
D00FE,00FF (Return)
DDT zeigt Ihnen:
00FE 07 01 ...
Daraus ist zu entnehmen, daß auf dem Stack die Rückkehradresse 0107 eingetragen wurde. Das ist in der Tat die Adresse innerhalb unseres Beispielprogramms, die dem BDOS-Call folgt.
Führen Sie den nächsten Trace aus. Das Resultat sollte sein (Bild 5):
Diese Befehle sind Bestandteil der BDOS-Initialisation. Die Adressen der Sprungbefehle können von der im Beispiel erwähnten abweichen, je nach Konfiguration des geladenen CP/M.
Die folgenden drei Traces dienen der Rettung des Stack-Pointers. Der Trace danach sollte etwa so aussehen (Bild 6):
Wie unmißverständlich klar wird, hat DDT eine unbestimmte Zahl von Programmschritten übersprungen, denn der Program-Counter zeigt auf 95A7 und der nächstfolgende Program-Counter-Stand soll 0107 werden. Was war in der Zwischenzeit?
Wenn Sie sich den Trace genau ansehen, so stellen Sie fest, daß nach dem disassemblierten Befehl (JMP 9F06) der Buchstabe Q in der Zeile erscheint. Dieses Q ist das von unserem Beispielprogramm ausgegebene Q. So stellt DDT alle Konsolen-Ausgaben dar, das auszugebende Zeichen folgt nach dem disassemblierten Befehl in der Trace-Zeile. Die zwischen dem Aufruf der BDOS-Funktion und seiner Ausführung liegenden Programmschritte zeigt DDT nicht! Diesen Bereich durchläuft DDT in Echtzeit, d.h. ohne Trace. Aus diesem Grund erscheint das Q, ohne daß die einzelnen Programmschritte nachvollziehbar sind. DDT tut das übrigens nicht, um dem Anwender zu verheimlichen, was im BDOS-Teil des Programms vorgeht, sondern weil hier in vielen Fällen Routinen laufen, die nicht unterbrochen werden dürfen, weil sie zeitkritisch sind. Der nächste Trace führt uns "nach Hause" (Bild 7):
An dieser Stelle tun Sie gut daran, mit dem Tracen aufzuhören! Offensichtlich würde der RET-Befehl auf die Adresse 020E führen, ein Bereich, den wir nicht definiert haben. Unser RET-Befehl sollte aus dem Programm hinaus auf die Betriebssystem-Ebene führen. Unter DDT sind diese Gesetze nicht gültig! Damit ist der Test unseres Programms beendet.

Wie es weitergeht:
Mit dieser Beschreibung des Programms DDT sind die Gratis-Tools grundsätzlich vorgestellt. In der nächsten Ausgabe soll anhand eines weiteren Beispiels die Praxis im Umgang mit den Tools vertieft werden. Bei dieser Gelegenheit lernen Sie noch einiges über BDOS-Calls.

(Dipl. Ing. H. Bruhns)

Bild 1
```
C0Z0M0E0I0 A=00 B=0000 D=0000 H=0000 S=0100 P=0100  MVI   C,02*0102
```
Bild 2
```
C0Z0M0E0I0 A=00 B=0002 D=0000 H=0000 S=0100 P=0102  MVI   E,51*0104
```
Bild 3
```
C0Z0M0E0I0 A=00 B=0002 D=0051 H=0000 S=0100 P=0104  CALL  0005*0005
```
Bild 4
```
C0Z0M0E0I0 A=00 B=0002 D=0051 H=0000 S=00FE P=0005  JMP   8F00*8F00
```
Bild 5
```
C0Z0M0E0I0 A=00 B=0002 D=0051 H=0000 S=00FE P=8F00  JMP  95A2*95A2
```
Bild 6
```
C0Z0M0E0I0 A=00 B=0002 D=0051 H=0000 S=00FE P=95A7  JMP  9F06Q*0107
```
Bild 7
```
C0Z1M0E1I0 A=00 B=0051 D=C4A7 H=0000 S=0100 P=0107  RET   *020E
```



8 KByte mehr Speicherplatz

auch im CP/M-Format

für 464-664-6128



Der auf den Schneider 3"-Disketten zur Verfügung stehende Speicherplatz ist nicht gerade üppig bemessen; speziell bei Verwendung von CP/M 3.0 sind Anwender immer auf der Suche nach weiteren Bytes für Speicherplatz, und wichtige Systemprogramme sind leider nur sehr schnell gelöscht. Das hier vorgestellte Verfahren stellt nun 8 KByte mehr Speicherplatz auf der normalen Schneider 3"-Diskette zur Verfügung. Dazu ist es lediglich erforderlich, die Diskette mit einem geeigneten Programm bis zur 42. Spur zu formatieren. Anschließend wird das gewünschte Programm mit **'PIP'** auf die neu formatierte Diskette übertragen. Wenn man sich danach mit einem Diskettenmonitor die Spur 2 und den Sektor 1 dieser Spur ansieht, wird man feststellen, daß das mit **'PIP'** übertragene Programm in diesem Sektor steht, einem Teil des Directorys für Disketten im Systemformat, das sich ja vom ersten bis zum vierten Sektor der Spur 2 erstreckt. Dabei stellt das erste Byte die Usernummer (0 - 15 hex) dar, darauf folgen Dateiname und Dateikennung. Das 16. Byte des Directoryeintrages kennzeichnet die Länge des Programmes. Das 17. und die darauf folgenden Bytes kennzeichnen die Blöcke (jeweils 1 KByte) auf der Diskette, in denen das Programm abgespeichert ist. Für die Umrechnung der Blöcke in Spur und Sektor ergibt sich folgende Beziehung, die leicht auch in ein entsprechendes Umrechnungsprogramm umgesetzt werden kann:
Die dem Block (Eingabe in Hex) entsprechende Spur ergibt sich nach der Formel:

Spur=INT(Block*2/Blockanzahl)+Systemspuren

Der entsprechende Sektor läßt sich berechnen nach:

Sektor=Block*2-Spur*Blockanzahl+ Systemspuren*Blockanzahl+1

Dabei ist der Wert für den Block auf der Diskette (1 KByte, entsprechend zwei Sektoren) in Hex einzugeben, die Werte für die Blockanzahl und die Systemspuren richten sich nach dem verwendeten Format. Beim uns hier interessierenden System- bzw. Vendorformat, das ja nur ein Systemformat ohne überspieltes System ist, müssen wir die Werte **2** für die Anzahl der Systemspuren einsetzen, und **9** für die Anzahl der möglichen Blöcke auf einer Spur. Mit diesen beiden Formeln lassen sich jetzt an Hand der im Directoryeintrag gefundenen Werte die Spuren und Sektoren berechnen, in denen unser Programm zu finden ist. Jetzt gilt es nur noch, diese Sektoren auf die beim Formatieren bis zur 42. Spur neu gewonnenen Sektoren zu übertragen. Leider setzt das Betriebssystem des CPC dem Versuch, diese Sektoren sinnvoll zu nutzen, einigen Widerstand entgegen. Aber wenn man einige Feinheiten beachtet, lassen sie sich letztendlich dennoch nutzen, ohne daß man in der Benutzung der Disketten Unterschiede zum normalen Verfahren bemerkt. So können (bis jetzt) von den zusätzlich gewonnenen 27 Sektoren leider nur 16 benutzt werden. Wenn man versucht mehr auszunutzen, reagiert der CPC recht allergisch mit allen nur denkbaren Fehlermeldungen vom AMSDOS. Darüber hinaus konnten die Sektoren immer nur paarweise belegt werden, an sich auch verständlich, wenn man bedenkt, daß immer nur 1K-Blöcke (1024 Bytes) abgespeichert werden, und daß ein Sektor 512 Bytes Inhalt hat. So konnten nur die Sektoren 1 - 8 auf den Spuren 41 und 43 verwendet werden, insgesamt also 8 KByte. Ob man hier ein oder mehrere Programme unterbrachte, blieb sich gleich, auf alle Fälle war genügend Platz für einen Diskmonitor, das DISCKIT-Programm oder anderes.
Um das gewünschte Programm an Ort und Stelle zu bringen, geht man folgendermaßen vor:
Man liest das Programm sektorweise in den Rechner ein, um direkt anschließend den Sektor an der gewünschten Stelle, z.B. Spur 40, Sektor 1, wieder abzuspeichern. Danach kommt der nächste Sektor an die Reihe, der dann auf Spur 40 als Sektor 2 abgespeichert wird usw. bis zum Sektor 8 (falls das Programm so lang sein sollte). Längere Programme sollten dann, beginnend im Sektor 1 der Spur 42, weiter abgespeichert werden. Ebenso könnte dort ein zweites Programm beginnen, das in CP/M, unabhängig vom zuerst abgespeicherten, gestartet werden kann. Um diese Arbeit zu erleichtern, ist im Sonderheft 2, CPC International ein Programm erschienen, das diesen Transfer auf die gewünschten Spuren und Sektoren erledigt. Anschließend muß nur noch im Inhaltsverzeichnis (Directory) der Diskette statt der ursprünglichen Blocks mit der Angabe der Spuren und Sektoren, in denen das Programm gespeichert war, der Block oder möglicherweise auch die Blöcke mit den neuen Speicherstellen eingetragen werden, damit das Betriebssystem die Programme auch wiederfinden kann. Im Prinzip ist die Darstellung des Verfahrens, zusätzlichen Speicherplatz auf der 3"-Diskette (übrigens funktioniert die Sache auch mit 3,5"- und 5 ¼"-Laufwerken) identisch mit demjenigen, das bereits im Sonderheft 2 für das Datenformat beschrieben worden ist. Jedoch aufgrund der Tatsache, daß gleiche Hexwerte im Directoryeintrag bei verschiedenen Formaten unterschiedliche Werte für Spur bzw. Sektor ergeben und auch beim Systemformat mit der vorgegebenen Tabelle gearbeitet wurde, führte zu dem Trugschluß, daß im CP/M-Format eine derartige Nutzung nicht möglich sei. In der Tat funktioniert dieses Verfahren, wie man sehen wird, unter Berücksichtigung der veränderten Voraussetzungen auch im CP/M-Format einwandfrei. Das gilt übrigens für CP/M 2.2 und CP/M 3.0. Auch alle anderen, im Sonderheft bereits angeführten, Chrakteristika dieses Verfahrens bleiben hier erhalten, so z.B. die Tatsache, daß die Programme von DISCKIT nicht überschrieben werden usw.

Transfer von Programmen auf die Spuren 40/42

Das folgende Programm ermöglicht es Ihnen, die zusätzlichen Spuren Ihrer Diskette zu nutzen.
Die Schneider Diskettenstation erlaubt ein Formatieren bis zur 43. Spur. Auf diesen, über die 40. Spur hinaus, zusätzlich gewonnenen Spuren können mit Hilfe bestimmter Utilities auch Programme abgespeichert werden. Zum Einlesen benötigt man dann jedoch wieder spezielle Programme, die die auf diesen Spuren abgespeicherten Programme einlesen. Das ist leider etwas umständlich, da

zwei verschiedene Utilities ständig greifbar sein müssen. Diesen Nachteil vermeidet das hier geschilderte Verfahren, da die, mit Hilfe dieses Programmes auf zusätzlich gewonnenen Spuren, überspielten Programme sich wie gewohnt von Basic oder unter CP/M aufrufen lassen, jedoch keinen Speicherplatz innerhalb des normalerweise auf der Diskette zur Verfügung stehenden Raumes beanspruchen, da sie auf den Spuren 40 und 42 untergebracht werden. Lediglich für den Transfer der Programme auf die Spuren 40/42 wird einmalig das hier gezeigte Programm benötigt.

Programmbeschreibung:
Für das Einlesen der Programme von der Diskette, das bei diesem Programm sektorweise erfolgt, wird, ebenso wie für das Schreiben der Sektoren, z.B. auf Spur 40, ein kleines Maschinenprogramm benötigt, das zunächst einmal mit Hilfe eines Ladeprogrammes in einem Speicherbereich des Rechners geschützt untergebracht wird. Danach erwartet der Rechner die Mitteilung, ob es sich bei der Diskette um eine im Data-Format oder im CP/M-Format formatierte Diskette handelt. Danach stellt der Rechner fest, welche Sektoren auf den Spuren 40/42 bereits belegt sind, und zeigt den noch zur Verfügung stehenden Speicherplatz an (maximal 8 KByte). Anschließend zeigt er mit dem 'CAT'-Befehl alle auf der Diskette gespeicherten Programme und erwartet die Eingabe des Programmnamens von dem Programm, das auf die Spuren 40/42 transferiert werden soll. Dabei ist der Dateiname mit Dateikennung (Extension) einzugeben (z.B. DISCKIT2.COM). Daraufhin durchsucht er das Directory, bis er einen Eintrag mit dem Namen des zu übertragenden Programmes gefunden hat, ermittelt die Programmlänge und die Position des Programmes auf der Diskette. Dann wird das Programm Sektor für Sektor in den Rechner eingelesen und Sektor für Sektor auch wieder auf den Spuren 40/42 abgespeichert. Nachdem das geschehen ist, wird es durch Eintrag im Directory zum "Read-only-File" erklärt. Die geänderte Position des Programmes wird im Directory eingetragen und der ursprünglich vom Programm eingenommene Platz steht für die Diskettenverwaltung wieder zur Verfügung. Anschließend fragt das Programm, ob noch ein weiteres Programm übertragen werden soll. Wird diese Frage mit "j" beantwortet, beginnt der gesamte Ablauf von vorne, wird die Frage mit "n" beantwortet, wird ein Kaltstart durchgeführt.

Die Bedienung des Programms ist ganz einfach und erklärt sich im wesentlichen von selbst. Wichtig ist jedoch, daß die Diskette, die mit Hilfe dieses Programmes bearbeitet werden soll, vorher mit einem Formatierprogramm bis zur Spur 42 formatiert sein muß. Dafür eignet sich das im Sonderheft 2 geschilderte DISC-BASIC V3.0 oder auch andere Programme, die ein Formatieren bis zur Spur 42 erlauben. Auch muß unbedingt beim Eingeben des Programmnamens die Dateikennung mit eingegeben werden, da es sonst zu Fehlermeldungen kommt, weil das Programm nicht in der Lage ist, die Programme im Directory zu finden. Darüber hinaus sollte auch die Eingabe des Diskettenformates stimmen. Sollte es zu Disc-Errors kommen, so liegt das gewöhnlich an der falschen Eingabe des Formates. Wenn all diese Hinweise beachtet werden, so werden auf Ihren Disketten von jetzt an 8 KByte mehr an Speicherplatz zur Verfügung stehen, genügend Platz, um zum Beispiel unter CP/M das sehr oft benötigte Programm "DISCKIT" o.ä. ständig zur Verfügung zu haben, oder im Basic Programme wie Maschinensprachemonitore, Formatierprogramme usw.

(W. Wantia)

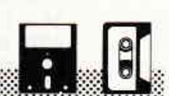
i$=chr$(24)	Inversdarstellung
a$	Tastatureingabe
n$, name$	Name des zu transferierenden Programms
fuell$	Auffüllung name$ auf 12 Zeichen
drive	aktuelles Laufwerk
Spur	aktuelle Spur
Sektor	aktueller Sektor
36864	1. Byte im Sektorpuffer
37406	letztes Byte im Puffer
block	Anfang des Programms in Hex (Directoryeintrag)
sysp	Anzahl Systemspuren bei gewähltem Format
n	Anfang des Programms (Sektorpuffer)
a	Programmlänge
ba	Blockanzahl 1KByte-Blöcke
b	10. Byte Directoryeintrag
d(c)	Position der übertragenen Sektoren auf Diskette
f	Formatoffset
t	Prüfvariable
z	Zähler Sektoranzahl
i,j,k	Laufvariable

```
10 '                                          [117]
20 '-------------------------------------Prog [3077]
ramm-Transfer auf Spur 40/42
30 '                                          [117]
40 '                                   by W [951]
.Wantia
80 '                                          [117]
90 MODE 2:GOSUB 880                           [1102]
100 CLS:i$=CHR$(24)                           [835]
110 '                                         [117]
120 '-------------------------------Initia [1706]
lisierung
130 '                                         [117]
140 LOCATE 27,3:PRINT i$" Initialisierung  [2517]
"i$;
150 LOCATE 10,6:PRINT"Transfer im "i$"C"i$ [9738]
"P/M- oder "i$"D"i$"ata-Format";:INPUT a$:
a$=UPPER$(a$):IF LEN(a$)>1 THEN PRINT CHR$
(7):GOTO 150
160 IF a$="C" THEN f=64:GOTO 220              [1361]
170 IF a$="D" THEN f=192:GOTO 220             [2030]
180 IF a$<>"C" OR a$<>"D" OR a$="" THEN 15 [2497]
0
190 '                                         [117]
200 '-------------------------------Festst [3048]
ellen freien Speicherraums
210 '                                         [117]
220 drive(0)=0:spur(0)=40:sector(0)=1:GOSU [2451]
B 1060
230 IF PEEK(36864)=229 THEN ba=0:GOTO 360  [2465]
240 sector(0)=sector(0)+2:GOSUB 1060       [2086]
250 IF PEEK(36864)=229 THEN ba=ba+1:GOTO 3 [1570]
60
260 IF PEEK(36864)<>229 THEN ba=ba+1:IF ba [2120]
<3 THEN 240
270 spur(0)=42:sector(0)=1:GOSUB 1060      [2015]
280 IF PEEK(36864)=229 THEN ba=ba+1:GOTO 3 [1570]
60
290 IF PEEK(36864)<>229 THEN ba=ba+1       [1390]
300 sector(0)=sector(0)+2:GOSUB 1060       [2086]
310 IF PEEK(36864)=229 THEN ba=ba+1:GOTO 3 [1570]
60
320 IF PEEK(36864)<>229 THEN ba=ba+1:IF ba [1864]
<8 THEN 300
330 '                                         [117]
340 '-------------------------------Merke  [2071]
Programmname
350 '                                         [117]
360 CLS:CAT                                   [140]
370 LOCATE 10,22:PRINT"(Noch "8-ba"K-Byte  [6170]
zur Verfuegung)";:ba=8-ba
380 LOCATE 1,20:INPUT"Welches Programm sol [8716]
l uebertragen werden ( Programmname.EXT )"
;name$
390 IF LEN(name$)<12 THEN Fuell=12-LEN(nam [10681]
e$):FOR i=1 TO fuell:fuell$=fuell$+CHR$(32
```

```
):NEXT:name$=MID$(name$,1,LEN(name$)-4)+fu
ell$+MID$(name$,LEN(name$)-3,4)
400 a$=RIGHT$(name$,4):IF LEFT$(a$,1)<>CHR [7671]
$(46) THEN LOCATE 30,18:PRINT i$" Eingabef
ehler ! "i$:FOR k=1 TO 2000:NEXT:GOTO 200
410 IF LEN(name$)=12 THEN name$=MID$(name$ [3942]
,1,8)+RIGHT$(name$,3)
420 '                                      [117]
430 '--------------------------------Lesen [2751]
 Directory
440 '                                      [117]
450 drive(0)=0:drive(1)=0                  [771]
460 IF f=192 THEN Spur(0)=0:sector(0)=1:GO [2156]
TO 480
470 IF f=64 THEN spur(0)=2:sector(0)=1     [1822]
480 GOSUB 1060:GOSUB 1440                  [1076]
490 IF NOT n$=UPPER$(name$) THEN sector(0) [3477]
=2:GOSUB 1060:GOSUB 1440
500 IF NOT n$=UPPER$(name$) THEN sector(0) [3596]
=3:GOSUB 1060:GOSUB 1440
510 IF NOT n$=UPPER$(name$) THEN sector(0) [3434]
=4:GOSUB 1060:GOSUB 1440
520 IF n$<>UPPER$(name$) THEN CLS:LOCATE 1 [5322]
0,3:PRINT i$" Programm nicht gefunden !"i$
:FOR i= 1 TO 2000:NEXT:GOTO 840
530 GOSUB 1520                             [907]
540 IF a>16 OR INT(a/2)>ba THEN CLS:LOCATE [5812]
 10,3:PRINT i$"Fehler ! Programm zu lang !
"i$:FOR i=1 TO 2000:NEXT:GOTO 840
550 '                                      [117]
560 '-----------------------------Uebertr [3575]
agen des Programms
570 '                                      [117]
580 CLS:spur(0)=spur:sector(0)=sector:LOCA [10942]
TE 10,1:PRINT"Lese      : Spur";spur(0);:LOC
ATE 40,1:PRINT"Sector";sector(0);:GOSUB 10
60
590 IF ba=8 THEN spur(1)=40:sector(1)=1:LO [6976]
CATE 10,3:PRINT"Schreibe: Spur";spur(1);:L
OCATE 40,3:PRINT"Sector";sector(1);:GOSUB
960
600 IF ba<>8 THEN GOSUB 1640:LOCATE 10,3:P [7242]
RINT"Schreibe: Spur";spur(1);:LOCATE 40,3:
PRINT"Sector";sector(1);:GOSUB 960
610 GOSUB 1140:GOSUB 1060                  [1120]
620 IF z=a THEN CLS:LOCATE 10,3:PRINT" Pro [7072]
gramm uebertragen. Es wird vor Ueberschrei
ben geschuetzt":GOTO 670
630 GOSUB 1210:GOSUB 960:GOTO 610          [1497]
640 '                                      [117]
650 '--------------------------------Uebers [2033]
chreibschutz
660 '                                      [117]
670 IF f=192 THEN Spur(0)=0:sector(0)=1:GO [2146]
TO 690
680 IF f=64 THEN spur(0)=2:sector(0)=1     [1822]
690 GOSUB 1060:GOSUB 1440                  [1076]
700 IF NOT n$=UPPER$(name$) THEN sector(0) [3477]
=2:GOSUB 1060:GOSUB 1440
710 IF NOT n$=UPPER$(name$) THEN sector(0) [3596]
=3:GOSUB 1060:GOSUB 1440
720 IF NOT n$=UPPER$(name$) THEN sector(0) [3434]
=4:GOSUB 1060:GOSUB 1440
730 GOSUB 1520                             [907]
740 b=n-7:c=PEEK(b):POKE b,c+128           [2410]
750 '                                      [117]
760 '-------------------------------Direct [3312]
oryeintrag aendern
770 '                                      [117]
780 IF f=192 THEN d(1)=&B4:d(2)=&B5:d(3)=& [3723]
B6:d(4)=&B7:d(5)=&BD:d(6)=&BE:d(7)=&BF:d(8
)=&C0
790 IF f=64 THEN d(1)=&AB:d(2)=&AC:d(3)=&A [4157]
D:d(4)=&AE:d(5)=&B4:d(6)=&B5:d(7)=&B6:d(8)
=&B7
800 t=a/2:IF INT(a/2)<t THEN a=INT(a+1)/2  [2980]
ELSE a=INT(a/2)
810 IF ba=8 THEN FOR c=1 TO a:POKE n,d(c): [2828]
n=n+1:NEXT
820 IF ba<>8 THEN FOR c=8-ba+1 TO 8-ba+a:P [3907]
OKE n,d(c):n=n+1:NEXT
830 sector(1)=sector(0):spur(1)=spur(0):GO [4351]
SUB 970
840 z=0:n=0:a=0:ba=0:b=0:t=0:j=0:i=0:k=0:n [5008]
ame$="":fuell=0:fuell$=""
850 CLS:LOCATE 1,3:PRINT"Soll noch ein Pro [6356]
gramm uebertragen werden ? (j/n)";:INPUT a
$:a$=UPPER$(a$)
860 IF a$="J" THEN 190                     [900]
870 IF a$="N" OR a$<>"J" OR a$<>"N" THEN C [1983]
ALL &0
880 '                                      [117]
890 '-------------------------------M-Code [2026]
 Initialisierung
900 '                                      [117]
910 CLS:PRINT"Bitte Geduld, M-Code wird ge [3413]
speichert"
920 GOSUB 1250                             [873]
930 MEMORY &9000-1                         [548]
940 FOR I=&9000 TO &9000+512:POKE I,0:NEXT [2508]
950 RETURN                                 [555]
960 '                                      [117]
970 '----------------------------------Rou [3165]
tine Sector speichern
980 '                                      [117]
990 POKE &A00E,DRIVE(1)                    [1388]
1000 POKE &A010,SPUR(1)                    [872]
1010 POKE &A012,(SECTOR(1) OR f)           [1405]
1020 POKE &A01A,&85                        [768]
1030 CALL &A000                            [637]
1040 POKE &A01A,&84                        [765]
1050 RETURN                                [555]
1060 '                                     [117]
1070 '----------------------------------Ro [3161]
utine Sector laden
1080 '                                     [117]
1090 POKE &A00E,DRIVE(0)                   [1410]
1100 POKE &A010,SPUR(0)                    [870]
1110 POKE &A012,(SECTOR(0) OR f)           [1437]
1120 CALL &A000                            [637]
1130 RETURN                                [555]
1140 '                                     [117]
1150 '----------------------------------Za [2615]
ehlschleifen
1160 '                                     [117]
1170 sector(0)=sector(0)+1:z=z+1           [2232]
1180 IF sector(0)>9 THEN sector(0)=1:spur( [3173]
0)=spur(0)+1
1190 LOCATE 10,1:PRINT"Lese      : Spur";spu [4131]
r(0);:LOCATE 40,1:PRINT"Sector";sector(0);
1200 RETURN                                [555]
1210 sector(1)=sector(1)+1                 [2466]
1220 IF sector(1)>8 THEN sector(1)=1:spur( [3522]
1)=spur(1)+2
1230 LOCATE 10,3:PRINT"Schreibe: Spur";spu [6875]
r(1);:LOCATE 40,3:PRINT"Sector";sector(1);
1240 RETURN                                [555]
1250 '                                     [117]
1260 '--------------------------------MC-C [3346]
ode Lader
1270 '                                     [117]
1280 DATA 21,1A,A0,CD,D4,BC,22,1B,A0,79,32 [3468]
,1D,A0,1E,00,16,= 1457
1290 DATA 00,0E,00,21,00,90,DF,1B,A0,C9,84 [2089]
,00,00,00,00,00,= 934
1300 '                                     [117]
1310 MEMORY &9FFF                          [134]
1320 zeile = 10000 : schritt = 10          [1008]
1330 adr = &A000 : last = &A01F            [1198]
1340 FOR i=1 TO 16                         [317]
1350      READ byte$                       [603]
1360      POKE adr,VAL("&"+byte$)          [745]
1370      sum = sum + PEEK(adr)            [1499]
1380      adr = adr + 1                    [392]
1390 NEXT                                  [350]
1400 READ checksum$ : checksum = VAL(MID$( [3979]
checksum$,3))
1410 IF sum <> checksum THEN PRINT "Fehler [3386]
 in Zeile";zeile
1420 IF adr < last THEN sum = 0 : zeile =  [2286]
zeile + schritt : GOTO 1340
1430 RETURN                                [555]
1440 '                                     [117]
1450 '-------------------------------Finde [3881]
Programm in Directory
1460 '                                     [117]
1470 n$="":i=36865:a=i+10:FOR j=i TO a:n$= [2617]
n$+CHR$(PEEK(j)):NEXT
1480 IF n$=UPPER$(name$) THEN RETURN       [3114]
1490 i=i+32:n$="":a=i+10:FOR j=i TO a:n$=n [3182]
$+CHR$(PEEK(j)):NEXT
1500 IF n$=UPPER$(name$) THEN RETURN ELSE  [5578]
IF j<37406 THEN GOTO 1490
1510 FOR j=36864 TO 37406:POKE j,0:NEXT:a= [2332]
0:i=0:j=0:RETURN
1520 '                                     [117]
1530 '-------------------------------Berech [4876]
nen der Programmlaenge
```

```
1540 '                                      [117]
1550 n=a+5: a=a+4:a=PEEK(a)*128:t=a/512     [1971]
1560 IF INT(a/512)<t THEN a=INT(a/512)+1 E  [3109]
LSE a=INT(a/512)
1570 '                                      [117]
1580 '------------------------------Berech  [5312]
nen des Programmanfangs
1590 '                                      [117]
1600 block=PEEK(n):IF f=192 THEN syssp=0 E  [4798]
LSE syssp=2
1610 spur=INT(block*2/9)+syssp              [1496]
1620 sector=block*2-spur*9+syssp*9+1        [2661]
1630 RETURN                                 [555]
1640 '                                      [117]
1650 '------------------------------Progra  [4360]
mmanfang weitere Programme
1660 '                                      [117]
1670 IF ba=1 THEN spur(1)=42:sector(1)=7:R  [2763]
ETURN
1680 IF ba=2 THEN spur(1)=42:sector(1)=5:R  [2316]
ETURN
1690 IF ba=3 THEN spur(1)=42:sector(1)=3:R  [1625]
ETURN
1700 IF ba=4 THEN spur(1)=42:sector(1)=1:R  [2550]
ETURN
1710 IF ba=5 THEN spur(1)=40:sector(1)=7:R  [2127]
ETURN
1720 IF ba=6 THEN spur(1)=40:sector(1)=5:R  [3603]
ETURN
1730 IF ba=7 THEN spur(1)=40:sector(1)=3:R  [2228]
ETURN
```

CPC-Listingservice für 464-664-6128



Wie in jedem Heft, finden Sie auch hier wieder den Listingservice, der Ihnen helfen soll, die in unserem Magazin abgedruckte "Paperware" in "Software" für Ihren Rechner zu verwandeln. Neben den Piktogrammen für Kassetten-/Diskettenbetrieb und der Angabe der Lauffähigkeit finden Sie die Super-Eintipphilfe "Checksummer". Um den Checksummer benutzen zu können, genügt es, das Programm einmal aufzurufen; es generiert selbständig den Maschinencode, aktiviert den Checksummer, gibt eine Bereitschaftsmeldung aus und löscht sich anschließend selbst wieder. Ab jetzt wird jede Eingabe, nachdem man ENTER gedrückt hat, mit der Ausgabe einer Quersumme quittiert. Beim Eintippen von Programmlistings kann man nun durch den Vergleich der beiden in eckigen Klammern stehenden Zahlen feststellen, ob man sich vertippt hat und ggf. den Fehler sofort berichtigen.

Des weiteren hat man die Möglichkeit, mit dem RSX-Befehl CHECK,s,z ein Listing mit den dazugehörigen Checksummen zu erzeugen. Die Parameter s und z definieren den Ausgabekanal (s = 0 bis 9) und die Zeilennummer, ab der mit der Ausgabe begonnen werden soll; wird z nicht angegeben, so wird an dessen Stelle 1 angenommen.

Vor dem Druck sollte man mit WIDTH x die maximale Breite (x) einer Zeile festlegen. Mit Hilfe der RSX-Befehle ON und OFF kann der Cheksummer ein- bzw. ausgeschaltet werden.

Bei der Berechnung der Quersumme werden die Zeilennummer und Leerzeichen überlesen; zwischen Groß- und Kleinschreibung wird nur innerhalb zweier Anführungsstriche unterschieden, da ansonsten die Schreibweise von Befehlsworten und Variablennamen keinen Einfluß auf die Funktionstüchtigkeit eines Basicprogrammes hat.

Version 464 (Thomas Fippl)

```
100 REM *** CHECKSUM 464 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
```

```
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,98,aa,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 06,dd,11,3a,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a2
310 DATA c1,cd,a3,e7,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,3c,c4,e5,09,e3,cd,63
340 DATA e1,21,a4,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,96,f2,e3,cd,f6
370 DATA a5,cd,4e,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,4e,c3,18,f5
390 DATA 3a,24,ac,d6,08,47,7e,a7
400 DATA c8,cd,45,e1,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,8a,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,5c,c3,3e,5b,cd,5c,c3
520 DATA cd,79,ee,3e,5d,c3,5c,c3
```

Version 664

```
100 REM *** CHECKSUM 664 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 4c,cb,11,5b,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a9
310 DATA c1,cd,69,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,75,c4,e5,09,e3,cd,59
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,58,f3,e3,cd,f6
370 DATA a5,cd,9b,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,9b,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,22,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a3,c3,3e,5b,cd,a3,c3
520 DATA cd,49,ef,3e,5d,c3,a3,c3
```

Version 6128

```
100 REM *** CHECKSUM 6128 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 49,cb,11,5e,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a6
310 DATA c1,cd,64,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,72,c4,e5,09,e3,cd,54
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,53,f3,e3,cd,f6
370 DATA a5,cd,98,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,98,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,1d,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a0,c3,3e,5b,cd,a0,c3
520 DATA cd,44,ef,3e,5d,c3,a0,c3
```

Vario

WordStar-Modifikation

für 464-664-6128



Daß WordStar, aus dem Hause MicroPro, ein sehr anpassungsfähiges und variables Textsystem ist, braucht wohl kaum noch erwähnt zu werden. Inzwischen gibt es unzählige Änderungen und Verbesserungen an diesem System.
Als ich zum ersten Mal mit der Version für den Schneider CPC 464 arbeitete, war ich aber doch etwas unangenehm überrascht über die "ausgefransten" Kommentar- und Hilfstexte dieser WordStar-Version.
Bei näherer Betrachtung ergab sich folgender Sachverhalt:
Die Entwickler von WordStar waren von einem Bildschirm ausgegangen, der in der Lage war, Zeichen in zwei verschiedenen Helligkeitsstufen auf den Bildschirm zu schreiben. Dieses wurde in der vorliegenden Version durch normale und inverse Zeichen ersetzt.
Damit wollte ich mich aber nicht zufriedengeben. Ich beschloß, das Programm so abzuändern, daß auch auf dem Schneider mit zwei verschiedenen Schriftarten gearbeitet werden konnte.
Der folgende Text gibt nun Aufschluß darüber, wie es gemacht wird.
Fangen wir also an!

Was wird benötigt?
1. Eine WordStar-Version 3.0 (möglichst jene von Markt & Technik)
Sinn und Zweck:
Wir wollen ein Unterprogramm erstellen, welches im Ton-

puffer (ab #B61A) abgelegt werden soll (damit kein Speicherplatz im TPA verlorengeht), und uns zwei unterschiedliche Schriftarten zur Verfügung stellt.

Zum Programm:

Das Programm liegt als DATA-Lader vor. Dies ist die bequemste Art der Eingabe.

DATA-LADER

Hier nun der Lader für das 'VARIO'-Programm:

Version 464

```
300 '************************************** [1669]
****
310 '                                       [117]
320 '                                       [117]
330 '                                       [117]
340 '                                       [117]
350 DATA &01,&74,&00,&21,&14,&01,&11,&1A    [1516]
360 DATA &B6,&ED,&B0,&21,&76,&B6,&22,&D4    [1810]
370 DATA &BD,&C3,&00,&00                    [1008]
380 DATA &45,&4F,&21,&8C,&B6,&35,&C2,&2D    [1049]
390 DATA &B6,&34,&79,&FE,&1F,&C2,&35,&B6    [1548]
400 DATA &3E,&03,&77,&11,&09,&00,&2A,&01    [2035]
410 DATA &00,&19,&E9,&FE,&18,&C2,&2D,&B6    [1436]
420 DATA &3A,&8B,&B6,&FE,&01,&CA,&6A,&B6    [1963]
430 DATA &3E,&01,&32,&8B,&B6,&21,&50,&B6    [1507]
440 DATA &22,&D4,&BD,&3E,&18,&C9,&E5,&CD    [1131]
450 DATA &D3,&12,&EB,&E1,&CD,&64,&0B,&0E    [1727]
460 DATA &08,&41,&1A,&EB,&CB,&3F,&A6,&EB    [957]
470 DATA &77,&7C,&81,&67,&1C,&10,&F3,&C9    [1490]
480 DATA &21,&76,&B6,&22,&D4,&BD,&3E,&18    [1352]
490 DATA &32,&8B,&B6,&C9,&E5,&CD,&D3,&12    [1335]
500 DATA &EB,&E1,&CD,&64,&0B,&0E,&08,&41    [1635]
510 DATA &1A,&77,&7C,&81,&67,&1C,&10,&F8    [1641]
520 DATA &C9,&00,&01,&00,&00                [1123]
530 DATA &FF                                [109]
540 '                                       [117]
550 '                                       [117]
555 'Einleseroutine                         [2799]
557 '                                       [117]
560 dat=0:sz=0                              [85]
570 OPENOUT"VARIO.COM"                      [1191]
580 READ byte:dat=dat+1                     [956]
590 IF byte=&FF THEN CLOSEOUT:PRINT"Alles   [3233]
o.k.":END
600 sz=sz+byte                              [619]
610 PRINT#9,CHR$(byte);                     [977]
620 GOTO 580                                [448]
630 END                                     [110]
```

Version 664/6128

```
100 '                                       [117]
110 '                                       [117]
120 '                                       [117]
130 '                                       [117]
140 DATA &01,&6B,&01,&21,&14,&01,&11,&B6    [1611]
150 DATA &B2,&ED,&B0,&21,&F7,&B2,&22,&D4    [1820]
160 DATA &BD,&C3,&00,&00                    [1008]
170 DATA &FE,&18,&C2,&F3,&02,&3A,&0C,&B3    [1317]
180 DATA &FE,&01,&CA,&EB,&B2,&3E,&01,&32    [1481]
190 DATA &0C,&B3,&21,&D1,&B2,&22,&D4,&BD    [906]
200 DATA &3E,&18,&C9,&E5,&CD,&A5,&BB,&EB    [2314]
210 DATA &E1,&CD,&1A,&BC,&0E,&08,&41,&1A    [1847]
220 DATA &EB,&CB,&3F,&A6,&EB,&77,&7C,&81    [834]
230 DATA &67,&1C,&10,&F3,&C9,&21,&F7,&B2    [1727]
240 DATA &22,&D4,&BD,&3E,&18,&32,&0C,&B3    [1756]
250 DATA &C9,&E5,&CD,&A5,&BB,&EB,&E1,&CD    [1824]
260 DATA &1A,&BC,&0E,&08,&41,&1A,&77,&7C    [1732]
270 DATA &81,&67,&1C,&10,&F8,&C9,&00,&00    [1820]
280 DATA &ff                                [109]
290 dat=0:sz=0                              [85]
300 OPENOUT"vario2.com"                     [1347]
310  READ byte:dat=dat+1                    [956]
320  IF byte=&FF THEN CLOSEOUT:PRINT"Alles  [2671]
O.K.":END
330  sz=sz+byte                             [619]
340  PRINT#9,CHR$(byte);                    [977]
350 GOTO 310                                [510]
360 END                                     [110]
```

INSTALLATION

1. Gehen Sie zur Installation wie folgt vor:

a) Geben Sie im CP/M-Modus ein: DDT WS.COM und <RETURN> (sollte sich DDT nicht zusammen mit der WordStar-Datei auf einer Diskette befinden, so kopieren Sie es bitte noch auf die Diskette).
b) Wenn DDT bereit ist, geben Sie ein: A02C0 und <RETURN>
c) DDT meldet sich dann mit: 02C0
d) Geben Sie dann ein: (02C0) JMP B61A und <RETURN>
e) Danach meldet sich DDT mit: 02C3
f) Geben Sie dann ein: . und <RETURN>
g) Danach geben Sie ˆC und <RETURN> ein.
h) Es meldet sich CP/M mit A> wieder.
i) Geben Sie nun ein: SAVE 64 WS.COM

Damit ist die Anpassung geschafft. Überprüfen Sie jeden Schritt, bevor Sie ihn abschließen!

2. Nach dem Eintippen und Starten des "VARIO"-Programmes befindet sich auf der jeweiligen Diskette das Programm "VARIO.COM".

Wird dieses gestartet und danach das WordStar-Programm geladen, so erscheinen dessen Meldungen nun in halbfetter Schrift. Eingegebene Texte aber werden in normaler Schrift abgebildet.

Sollte bei Ihnen etwas anderes erscheinen, so überprüfen Sie noch einmal sorgfältig das Listing.

Hat alles geklappt? Nun, dann viel Spaß!

Weitere Hinweise und Tips:

1. Wollen Sie auch einzelne Buchstaben oder ganze Wörter des Hilfstextes in normaler Schrift erscheinen lassen, so müssen Sie nur das 7. Bit der entsprechenden Zeichen setzen. Für diese Arbeit eignet sich ein Disk-Monitor besonders gut.

2. Die "Halbfett-Schrift" wird auch zur Blockkennzeichnung und für "weiche" Bindestriche genutzt. Die Bindestriche sind dadurch aber nicht mehr besonders gut von den "harten" zu unterscheiden, deshalb bietet es sich an, das "-"-Zeichen durch ein "+"-Zeichen zu ersetzen (Patchadresse: #03B8).

3. Soll das "VARIO"-Programm sofort ausgeführt werden, wenn CP/M gestartet wird, so gehen Sie wie folgt vor:

a) Gehen Sie in den CP/M-Modus.
b) Geben Sie ein: SETUP und <RETURN> (sollte sich das Programm SETUP.COM nicht auf Ihrer Diskette befinden, so kopieren Sie es jetzt auf diese).
c) Beantworten Sie die erste Frage mit N.
d) Geben Sie ein: VARIOˆM und <RETURN>.
e) Beantworten Sie nun alle Fragen mit Y.

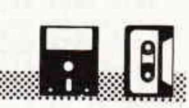
Nun wird jedesmal, wenn Sie CP/M aufrufen, als erstes das "VARIO"-Programm aufgerufen und abgearbeitet.

Wichtiger Hinweis:

Bevor Sie Ihre modifizierte WordStar-Version das erste Mal aufrufen, müssen Sie **immer** erst VARIO.COM aufrufen!!!

(Torsten Kirsch)

Seikosha Printer Improvement für 464-664-6128



Der Seikosha GP-550 A ist ein preisgünstiger Drucker, der innerhalb kürzester Zeit viele Freunde gefunden hat. Leider muß man für den niedrigen Preis auch einige Mängel

in Kauf nehmen. Damit sich Besitzer dieses Druckers nicht zurückgesetzt fühlen müssen, veröffentlichen wir in diesem Heft eine RSX-Erweiterung, die allen GP-550 A Anwendern neue Horizonte eröffnet, und so gut wie es eben softwaremäßig geht, die Mängel der Hardware ausbügelt. Angefangen von einer Hardcopy bis hin zu verschiedenen Schriftarten bietet das kurze MC-Programm alle Features, die bisher vermißt wurden.
Geben Sie einfach den Basiclader ein und starten ihn nach Anfertigung einer Sicherheitskopie mit RUN. Nachdem "READY" erscheint, geben Sie einfach nur: CALL &A000 und NEW ein. Danach stehen Ihnen folgende neue Basicbefehle zur Verfügung:

Befehle:

|HARDCOPY
Erzeugt einen Grafikausdruck von einer MODE 2-Screen. DIP-Schalter 4 muß sich in Stellung ON befinden.

|NORMAL,A$
Wie PRINT # 8-Befehl, aber in besserer Qualität und mit Zeichen über CHR$(127).

|INVERS,A$
Druckt den in A$ enthaltenen Text invers aus.

|BIG,A$
Der Text wird breit gedruckt.

|SMALL,A$
Der Text wird schmal gedruckt.

|KURSIV,A$
Der Text wird kursiv gedruckt.

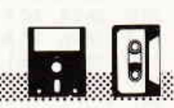
|MIXED,A$
Mischung zwischen Schmal- und Kursiv-Schrift.

Wichtig:
Vor der Anwendung der Befehle muß zuerst: SYMBOL AFTER 32 eingegeben werden.
Bei dem Befehl Hardcopy muß man den DIP-Schalter Nummer 4 auf "OFF" schalten.
Die Hardcopy kann durch das zweimalige Drücken der BREAK-TASTE unterbrochen werden. Es ist aber darauf zu achten, daß nach einer Unterbrechung sehr wahrscheinlich der Grafikmodus noch eingeschaltet ist.
(Sven Kottmann)

```
1000 '===
============================        [1959]
1010 '==  PRINTERIMPROVEMENT V2.0 ==       [1893]
1020 '== (c)1986 by Sven Kottmann ==       [1544]
1290 '                                     [117]
1300 zeile=1360:MEMORY &9FFF               [1083]
1310 FOR i=&A000 TO &A2DD STEP 15          [1208]
1320 s=0:FOR x=0 TO 14:READ a$:a=VAL("&"+a [3857]
$):POKE x+i,a:s=s+a:NEXT
1330 READ b$:IF VAL("&"+b$)<>s THEN PRINT" [7616]
ERROR IN DATAS in Zeile :";zeile:END ELSE
zeile=zeile+10
1340 NEXT                                  [350]
1350 PRINT "ALLES OK !":CALL &A000:END     [3106]
1360 DATA 21,00,A0,36,C9,3E,C3,32,04,AC,21 [3046]
,19,A0,22,05,04A4
1370 DATA AC,01,2D,A0,21,6C,A0,C3,D1,BC,7B [3534]
,FE,0D,C0,2A,0767
1380 DATA 34,AE,CD,3F,DD,7E,FE,7C,C0,2A,C2 [2424]
,B0,D1,D1,C9,098A
1390 DATA 44,A0,C3,70,A0,C3,A5,A1,C3,B9,A1 [3082]
,C3,C9,A1,C3,09CD
1400 DATA E9,A1,C3,F9,A1,C3,D9,A1,48,41,52 [2562]
,44,43,4F,50,0825
1410 DATA D9,49,4E,56,45,52,D3,42,49,C7,53 [2450]
,4D,41,4C,CC,067B
1420 DATA 4B,55,52,53,49,D6,4E,4F,52,4D,41 [3675]
,CC,4D,49,58,059B
1430 DATA 45,C4,00,00,00,00,00,21,00,C0,06 [2114]
,1C,CD,8D,A0,0406
1440 DATA CD,89,A1,10,F8,3E,0F,CD,75,A1,3E [2251]
,0D,CD,75,A1,075D
1450 DATA 3E,0A,CD,75,A1,C9,CD,5B,A1,C5,E5 [3383]
,E5,21,D7,A2,08E6
1460 DATA 36,00,E1,01,DF,01,CD,A7,A0,0B,79 [2937]
,B0,20,F8,E1,0739
1470 DATA C1,C9,C5,E5,3A,D7,A2,FE,00,28,1D [2528]
,FE,01,28,1E,076F
1480 DATA FE,02,28,1F,FE,03,28,20,FE,04,28 [2440]
,21,FE,05,28,0506
1490 DATA 22,FE,06,28,23,FE,07,28,24,C9,11 [2262]
,01,80,18,21,0456
1500 DATA 11,02,40,18,1C,11,03,20,18,17,11 [2455]
,04,10,18,12,0139
1510 DATA 11,05,08,18,0D,11,06,04,18,08,11 [2778]
,07,02,18,03,00B3
1520 DATA 11,00,01,0E,00,06,07,7E,A2,CD,16 [3120]
,A1,CD,26,BC,0480
1530 DATA 10,F6,79,CD,75,A1,3A,D7,A2,E1,FE [2690]
,07,CC,14,A1,087C
1540 DATA 7B,32,D7,A2,C1,C9,23,C9,D5,F5,78 [3970]
,FE,07,28,19,0824
1550 DATA FE,06,28,19,FE,05,28,19,FE,04,28 [1918]
,19,FE,03,28,04F5
1560 DATA 19,FE,02,28,19,FE,01,28,19,C9,16 [1847]
,01,18,16,16,03BE
1570 DATA 02,18,12,16,04,18,0E,16,08,18,0A [2602]
,16,10,18,06,00F0
1580 DATA 16,20,18,02,16,40,F1,B7,28,05,F5 [2277]
,7A,B1,4F,F1,05DB
1590 DATA D1,C9,3E,0A,CD,75,A1,3E,1B,CD,75 [2918]
,A1,3E,4B,CD,0757
1600 DATA 75,A1,3E,03,CD,75,A1,3E,5F,CD,75 [3787]
,A1,C9,CD,2E,077E
1610 DATA BD,F5,CD,00,B9,CD,3C,C4,CD,03,B9 [2994]
,F1,38,F0,CD,0974
1620 DATA 2B,BD,C9,C5,06,07,CD,26,BC,10,FB [2456]
,C1,3E,0D,CD,0716
1630 DATA 75,A1,C9,DD,66,01,DD,6E,00,46,23 [2749]
,7E,23,66,6F,064D
1640 DATA C9,CD,98,A1,AF,B8,28,07,7E,23,CD [2595]
,15,A2,10,F9,0793
1650 DATA 3E,0F,CD,75,A1,C9,CD,98,A1,AF,B8 [2682]
,28,F3,7E,23,0822
1660 DATA CD,2A,A2,10,F9,18,EA,CD,98,A1,AF [3605]
,B8,28,E3,7E,089A
1670 DATA 23,CD,39,A2,10,F9,18,DA,CD,98,A1 [3057]
,AF,B8,28,D3,082E
1680 DATA 7E,23,CD,48,A2,10,F9,18,CA,CD,98 [4045]
,A1,AF,B8,28,07D8
1690 DATA C3,7E,23,CD,5A,A2,10,F9,18,BA,CD [2120]
,98,A1,AF,B8,0875
1700 DATA 28,B3,7E,23,CD,6D,A2,10,F9,18,AA [1769]
,CD,74,A2,CD,07D3
1710 DATA 88,A2,21,CF,A2,06,08,C9,E5,C5,CD [2476]
,09,A2,7E,EE,0821
1720 DATA FF,77,23,10,F9,21,CF,A2,CD,9C,A2 [2379]
,C1,E1,C9,E5,098F
1730 DATA C5,CD,09,A2,7E,CB,3F,B6,77,23,10 [2702]
,F8,18,E8,E5,0802
1740 DATA C5,CD,09,A2,7E,CB,3F,A6,77,23,10 [2220]
,F8,18,D9,E5,07E3
1750 DATA C5,CD,09,A2,7E,CB,3F,A6,77,23,10 [2644]
,F8,21,CF,A2,079F
1760 DATA 18,05,E5,C5,CD,09,A2,23,23,23,23 [2991]
,06,04,7E,07,045A
1770 DATA 77,23,10,FA,18,B4,E5,C5,CD,09,A2 [1593]
,18,AD,CD,A5,07C9
1780 DATA BB,F5,D4,06,B9,11,CF,A2,01,08,00 [2643]
,ED,B0,F1,D4,0830
1790 DATA 09,B9,C9,3E,1B,CD,75,A1,3E,4B,CD [1543]
,75,A1,AF,CD,07AF
1800 DATA 75,A1,3E,08,CD,75,A1,C9,C5,E5,21 [2855]
,FC,A0,11,D8,0858
1810 DATA A2,01,03,00,E5,D5,C5,ED,B0,C1,D1 [1882]
,E1,E5,D5,C5,09B4
1820 DATA EB,21,DB,A2,ED,B0,21,CF,A2,AF,32 [2161]
,D7,A2,06,08,0820
1830 DATA CD,A7,A0,10,FB,C1,D1,E1,EB,ED,B0 [2995]
,E1,C1,C9,00,0A85
1840 DATA 00,00,00,00,00,00,00,00,00,00,00 [2187]
,23,00,00,00,0023
```

CPC Basic-Tuning

für 464-664-6128



Das Schneider-Basic ist schon recht schnell – und, ob Sie's glauben oder nicht, zwei POKE's reichen aus, um es noch schneller zu machen! Hier zwei Kurzprogramme im Vergleich:

10 FOR X=1 TO 10000:NEXT
Laufzeit: etwa 11 Sekunden.

10 POKE &B939,&C9
20 FOR X=1 TO 10000:NEXT
30 POKE &B939,&F3
Laufzeit: etwa 10 Sekunden.

Bitte beachten: Die Adresse &B939 gilt für den CPC 464; für den CPC 664/6128 muß &B941 eingesetzt werden.
Das Geheimnis dieser Steigerung? Sehr einfach: Ab Adresse &B939 beginnt die CPC-interne Interruptroutine, die 300 mal pro Sekunde aufgerufen wird und den normalen Ablauf unterbricht. C9 ist der Code für den Assembler-Befehl RET, der einen sofortigen Rücksprung bewirkt. Die Routine wird also nach dem POKE nicht mehr ausgeführt, und das Hauptprogramm läuft entsprechend schneller ab. Da aber gleichzeitig alle per Interrupt gesteuerten Funktionen lahmgelegt werden, sollte man diesen POKE nur mit Bedacht einsetzen.
Gibt man ihn im Direktmodus ein, ist der Rechner ab sofort nicht mehr ansprechbar, da die Tastaturabfrage unterbleibt. In diesem Fall hilft nur noch eins: aus- und wieder einschalten. Innerhalb eines Programms kann man jedoch durch POKE &B939,&F3 wieder den Normalzustand herstellen, bevor sich der Rechner mit Ready zurückmeldet.
Programmteile, die durch die beiden POKE's eingeschlossen sind, sollten also keine Tastaturabfrage beinhalten (INPUT, INKEY$). Weiterhin ist es nicht mehr möglich, die Bildschirmfarben mit dem INK-Befehl umzuschalten, und auch der Soundchip bleibt unversorgt. Ergänzen Sie das obige Programm durch die Zeile:
5 PRINT CHR$(7)
Zu hören ist ein penetranter Dauerton – der CPC hat nämlich keine Gelegenheit mehr, seinem Soundchip mitzuteilen, daß der Ton bereits zu Ende ist.
Trotz dieser Einschränkungen gibt es viele Fälle, in denen der Trick durchaus angebracht ist. Sortierroutinen in Dateiverwaltungsprogrammen kommen zum Beispiel problemlos ohne Sound und blinkende Farben aus, aber 10 % mehr Geschwindigkeit können sie sicherlich gut vertragen.

(Matthias Uphoff)

Farbtools für CP/M
für 464-664-6128



Beide Programme sind für User gedacht, die ihren Computer an einen Farbmonitor betreiben. Im Basic-Mode können die Farben leicht so eingestellt werden, daß Texte gut lesbar sind und auch ein längeres Arbeiten ohne zu große Ermüdung der Augen möglich ist. Nicht so einfach ist es, wenn man mit CP/M arbeitet. Mit dem Programm FARBE.COM lassen sich nun alle Farben für Hintergrund und Zeichen einstellen, und jeder Zeit, ohne CP/M zu verlassen, wieder ändern. Das Programm FARB.COM ist dagegen sehr kurz und setzt die Farben, wie sie im Assemblerlisting und im Basic-Programm nach ihrem eigenen Geschmack definiert worden sind. FARB.COM ist besonders geeignet, die Farben für ein Textprogramm einzustellen.

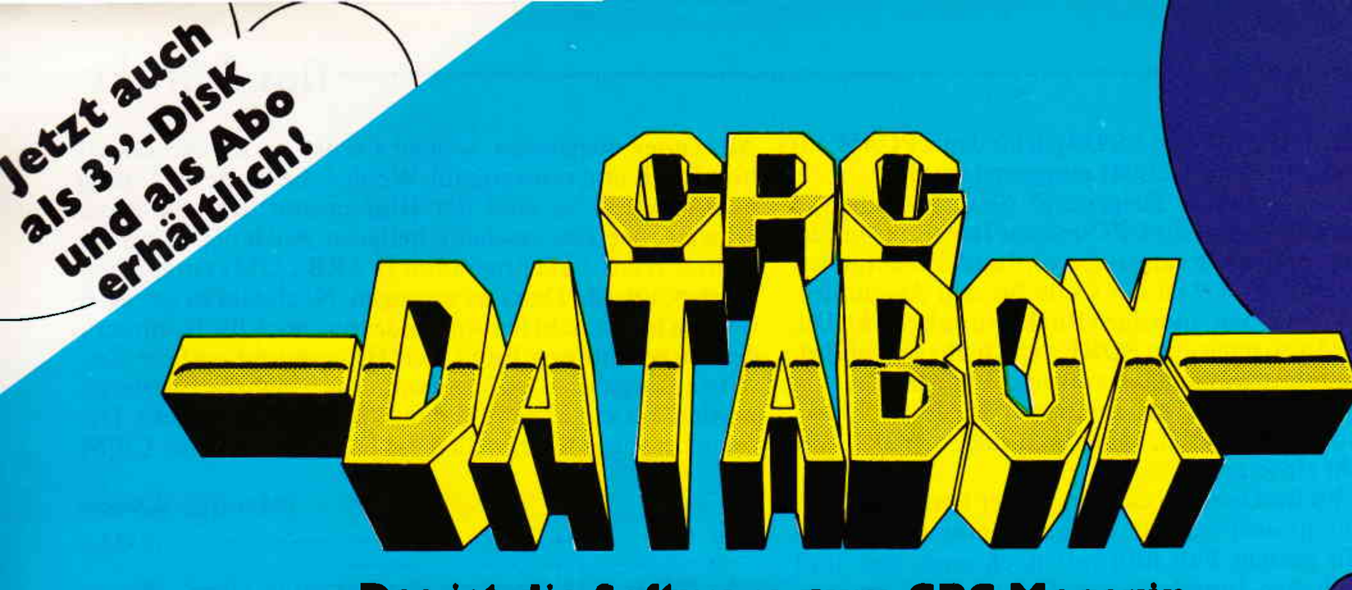
Beschreibung Programm FARBE.COM:
Die Datei FARBE.ASM ist das Assemblerlisting (8080) und kann mit dem CP/M-Editor ED, dem Assembler ASM und dem CP/M-Lader LOAD bearbeitet werden. Etwas mehr Arbeit bereitet das Eintippen des Basic-Ladeprogramms, das, nachdem es gestartet wird, automatisch die CP/M-Datei auf die Diskette schreibt. Das Programm verlangt vom Anwender nur wenige Eingaben, und ist auf dem Schirm ausreichend dokumentiert.

Beschreibung Programm FARB.COM:
Die Datei FARB.COM ist wieder das Assemblerlisting und kann wie oben verarbeitet werden. Es bleibt aber dem Anwender überlassen, welche Farben er für die Labels hint, zeich und rand eingibt. Werden die Werte im Listing übernommen, so sind der Hintergrund und der Rand schwarz, der Text erscheint hellgrün. Auch hier kann ein kleines Basic-Ladeprogramm (FARB.COM) eingegeben werden, um die Datei zu erzeugen. Nachdem es gestartet worden ist, braucht der Anwender nur noch die Nummern der Farben für den Rand, den Hintergrund und die Zeichen einzugeben. Das Programm kann mehrfach gestartet werden, bis die Farbwahl den Wünschen entspricht. Die zuletzt erzeugte Datei stellt diese Farben dann unter CP/M ein.

(Manfred Keuser)

```
100 REM --------------------------------- [3453]
------------------------------
110 REM Dieses Programm erzeugt eine COM-D [2281]
atei die unter CP/M aufgerufen
120 REM werden kann um die folgend gewaehl [3307]
ten Farben einzustellen.
130 REM --------------------------------- [3453]
------------------------------
140 REM                                   [272]
150 REM               Basiclader fuer Dat [929]
ei:farb.com
160 REM                        Vers. 1.1 - [693]
1986
170 REM                       M.Keuser - K [1594]
oeln
180 REM                                   [272]
190 MODE 2:x=6                            [988]
200 PRINT:PRINT " Waehlen Sie eine Farbe a [4853]
us der CPC Farbtabelle  (0...26)"
210 OPENOUT "farb.com"                    [1239]
220 FOR i=1 TO 25:READ b$                 [1282]
230 IF b$="hint" THEN wahl$="Hintergrund": [4202]
b$="0":GOSUB 320:INK 0,VAL("&"+b$)
240 IF b$="zeich" THEN wahl$="Zeichen":b$= [3529]
"12":GOSUB 320:INK 1,VAL("&"+b$)
250 IF b$="rand" THEN wahl$="Rand":b$="0": [3904]
GOSUB 320:BORDER VAL("&"+b$)
260 PRINT #9,CHR$(VAL("&"+b$));           [1033]
270 NEXT i:CLOSEOUT                       [667]
280 PRINT:PRINT " Die Datei FARB.COM ist a [6946]
uf der Diskette gespeichert !"
290 PRINT " Sie ruft unter CP/M die aktuel [4227]
len Farben auf !":PRINT:END
300 DATA 3E,00,06,hint,48,CD,32,BC,3E,01,0 [1615]
6,zeich,48,CD,32,BC
310 DATA 06,rand,48,CD,38,BC,C3,00,00     [1729]
320 x=x+1                                 [853]
330 LOCATE 7,x:PRINT "fuer ";wahl$        [1479]
340 LOCATE 32,x:PRINT "(nur ENTER setzt Fa [3728]
rbe";VAL("&"+b$);")"
350 LOCATE 25,x:INPUT "",farbe$           [2284]
360 IF VAL(farbe$)>26 THEN PRINT CHR$(7):G [2932]
OTO 330
370 IF farbe$="" THEN RETURN ELSE b$=HEX$( [3499]
VAL(farbe$)):RETURN
```

```
140 REM ********************************** [1898]
**
150 OPENOUT "farbe.com"                   [1239]
160 schritt=10:zeile=1000:summe=0         [1943]
170 FOR i=0 TO 15:READ b$                 [983]
180 IF b$="ende" THEN 250                 [664]
190 PRINT#9,CHR$(VAL("&"+b$));            [1033]
200 summe=summe+VAL("&"+b$)               [970]
210 NEXT                                  [350]
220 READ ck                               [574]
230 IF summe<>ck THEN PRINT "Fehler in Zei [3250]
le";zeile:GOTO 250
240 summe=0:zeile=zeile+schritt:GOTO 170  [2126]
250 CLOSEOUT:END                          [1352]
1000 DATA AF,CD,35,BC,78,3C,32,04,04,3E,01 [3945]
,CD,35,BC,78,3C, 1548
1010 DATA 32,05,04,CD,07,02,11,22,02,CD,0D [2564]
,02,11,EF,02,CD, 1009
1020 DATA 0D,02,CD,9C,BB,11,33,03,CD,0D,02 [2587]
,CD,9C,BB,21,0E, 1449
1030 DATA 1B,CD,75,BB,CD,13,02,FE,31,CA,49 [2470]
,01,FE,32,CA,1C, 1875
1040 DATA 02,FE,33,CA,62,01,C3,2E,01,3E,01 [2455]
,06,12,48,CD,32, 1264
```



- Das ist die Software zum CPC Magazin -
- Jeden Monat neu -





Inhalt der Databox zu Heft 6/86

Programm	464	664	6128
XBasic	●	●	●
XBasic.TXT	●	●	●
Transfer	●	●	●
Vario	●	●	●
Seiko Print	●		
Farbe	●	●	●
Farb	●	●	●
Pergo	●	●	●
Basic-Kurs	●	●	●
Mexico	●	●	●
Bonusprogramm	●	●	●

Für alle CPC's als Kassette und 3"-Diskette. Auch als Abo erhältlich!

DATABOX: – mehr als der übliche Softwareservice
DATABOX: – bringt ergänzend sämtliche Listings der jeweiligen Zeitschrift und alle Programmbeispiele auf Kassette oder auf 3"-Diskette.
DATABOX: – Programme sind, soweit systembedingt möglich, auf allen drei CPC-Modellen lauffähig. Einzelheiten entnehmen Sie bitte der nebenstehenden Aufstellung.
DATABOX: – erscheint jeden Monat und trägt das Titelbild des gleichzeitig erscheinenden Heftes.
DATABOX: – der Datenträger zum Schneider CPC International enthält außerdem jedesmal ein zusätzliches Bonusprogramm, das nicht im Heft abgedruckt ist.

Einzelbezug:

Einzelbezugspreise für DATABOX: Diskette 3" 24,– DM zuzüglich 3,– DM Porto/Verpackung (im Ausland zuzüglich 5,– DM Porto/Verpackung). Kassette 14,– DM zuzüglich Porto/Verpackung (im Ausland zuzüglich 5,– DM Porto/Verpackung).

Zahlungsweie: Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr (in das Ausland ist Nachnahme nicht möglich).

Schneider CPC International

Postfach 250, 3440 Eschwege

STOP – Bitte Bestellkarte benutzen –

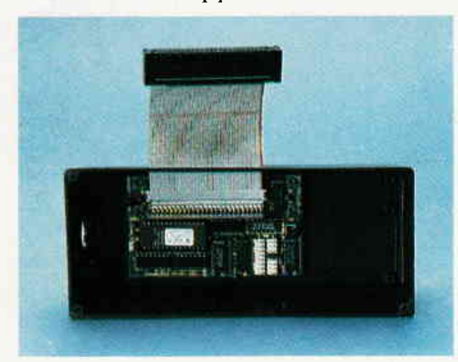
```
1050 DATA BC,3E,00,06,00,48,CD,32,BC,06,00 [2687]
,48,CD,38,BC,CD, 1503
1060 DATA 1C,02,CD,07,02,11,54,03,0E,09,CD [2511]
,05,00,3E,01,32, 694
1070 DATA 06,04,CD,9C,BB,21,09,0F,CD,75,BB [2428]
,11,F6,03,CD,0D, 1608
1080 DATA 02,CD,06,BB,FE,20,CA,CA,01,FE,73 [2537]
,CA,9B,01,FE,53, 2155
1090 DATA CA,9B,01,FE,0D,CA,19,02,C3,81,01 [2393]
,3A,06,04,FE,01, 1502
1100 DATA C2,B6,01,AF,32,06,04,21,09,0F,CD [2920]
,75,BB,11,E8,03, 1430
1110 DATA CD,0D,02,C3,81,01,3E,01,32,06,04 [1952]
,21,09,0F,CD,75, 1047
1120 DATA BB,11,F6,03,CD,0D,02,C3,81,01,3A [1947]
,06,04,FE,01,CA, 1523
1130 DATA F0,01,C3,D5,01,3A,04,04,47,3C,FE [3351]
,1B,C2,E0,01,AF, 1722
1140 DATA 32,04,04,48,AF,C5,CD,32,BC,C1,CD [3027]
,38,BC,C3,81,01, 1912
1150 DATA 3A,05,04,47,3C,FE,1B,C2,FB,01,AF [3621]
,32,05,04,48,3E, 1293
1160 DATA 01,CD,32,BC,C3,81,01,3E,02,CD,0E [2446]
,BC,C9,0E,09,CD, 1669
1170 DATA 05,00,C9,0E,01,CD,05,00,C9,CD,9C [3980]
,BB,CD,07,02,CD, 1599
1180 DATA 00,00,46,41,52,42,45,2E,43,4F,4D [3328]
,20,2D,20,56,65, 917
1190 DATA 72,73,69,6F,6E,20,31,2E,32,20,2D [3109]
,20,31,39,38,36, 1057
1200 DATA 0D,0A,2D,2D,2D,2D,2D,2D,2D,2D,2D [2089]
,2D,2D,2D,2D,2D, 653
1210 DATA 2D,2D,2D,2D,2D,2D,2D,2D,2D,2D,2D [2744]
,2D,2D,2D,2D,2D, 720
1220 DATA 0D,0A,0A,28,31,29,20,47,72,75,6E [2129]
,64,65,69,6E,73, 1138
1230 DATA 74,65,6C,6C,75,6E,67,20,2D,20,48 [2936]
,69,6E,74,65,72, 1490
1240 DATA 67,72,75,6E,64,20,3A,20,73,63,68 [3073]
,77,61,72,7A,0D, 1449
1250 DATA 0A,20,20,20,20,20,20,20,20,20,20 [2802]
,20,20,20,20,20, 490
1260 DATA 20,20,20,20,20,20,20,20,55,6D,72 [2125]
,61,6E,64,75,6E, 1098
1270 DATA 67,20,20,20,3A,20,73,63,68,77,61 [2553]
,72,7A,0D,0A,20, 1114
1280 DATA 20,20,20,20,20,20,20,20,20,20,20 [2072]
,20,20,20,20,20, 512
1290 DATA 20,20,20,20,20,20,5A,65,69,63,68 [2875]
,65,6E,20,20,20, 998
1300 DATA 20,20,3A,20,68,65,6C,6C,67,72,75 [1852]
,65,6E,20,24,0D, 1201
1310 DATA 0A,0A,28,32,29,20,42,69,6C,64,73 [2498]
,63,68,69,72,6D, 1208
1320 DATA 66,61,72,62,65,6E,20,75,65,62,65 [2302]
,72,6E,65,68,6D, 1609
1330 DATA 65,6E,0D,0A,0A,28,33,29,20,42,69 [3035]
,6C,64,73,63,68, 1105
1340 DATA 69,72,6D,66,61,72,62,65,6E,20,61 [2841]
,65,6E,64,65,72, 1605
1350 DATA 6E,20,24,0D,0A,0A,0A,0A,20,42,69 [3236]
,74,74,65,20,77, 918
1360 DATA 61,65,68,6C,65,6E,20,53,69,65,20 [2196]
,20,28,31,2E,2E, 1187
1370 DATA 33,29,20,24,46,61,72,62,65,69,6E [2704]
,73,74,65,6C,6C, 1403
1380 DATA 75,6E,67,0D,0A,2D,2D,2D,2D,2D,2D [3844]
,2D,2D,2D,2D,2D, 848
1390 DATA 2D,2D,2D,2D,0D,0A,0A,53,20,6F,64 [1855]
,65,72,20,73,20, 933
1400 DATA 20,2D,2D,3E,20,57,61,68,6C,20,56 [4038]
,6F,72,64,65,72, 1270
1410 DATA 2D,2F,48,69,6E,74,65,72,67,72,75 [2476]
,6E,64,0D,0A,4C, 1353
1420 DATA 65,65,72,74,61,73,74,65,20,2D,2D [3140]
,3E,20,46,61,72, 1358
1430 DATA 62,77,61,68,6C,0D,0A,45,6E,74,65 [2503]
,72,20,20,20,20, 1187
1440 DATA 20,2D,2D,3E,20,50,72,6F,67,72,61 [3137]
,6D,6D,20,62,65, 1284
1450 DATA 65,6E,64,65,6E,0D,0A,0A,0A,47,65 [3395]
,77,61,65,68,6C, 1266
1460 DATA 74,20,69,73,74,3A,20,24,20,48,69 [2910]
,6E,74,65,72,67, 1363
1470 DATA 72,75,6E,64,20,24,20,56,6F,72,64 [2086]
,65,72,67,72,75, 1501
1480 DATA 6E,64,20,24,00,00,00,00,00,00,00 [2390]
,00,00,00,00,00, 278
1490 DATA ende                              [481]
```

Das doppelte DOS



Das Disk-Operating-System der CPC Rechner ist ja an und für sich eine feine Sache. Noch viel schöner ist allerdings das VDOS der Vortex 5 1/4"-Station, meinen einige Leute.
„Schön und gut", hallt es von den 3"-Freaks zurück, „aber es läuft ja mit keiner bestehenden Software zusammen".
Der Kasus Knacksus in dieser Angelegenheit schien sich einfach nicht finden zu lassen. Bleibt also das Lager der CPC-Anwender in zwei Gruppen gespalten? Auf der einen Seite die Schneider-treuen Dreizöller und auf der anderen Seite die Fünf-Ein-Viertel-Vortexer?
NEIN!!!
Vortex, immer wieder für eine Überraschung gut, hat wohl festgestellt, daß eine angekündigte Festplatte und eine Speichererweiterung die Sensation und den Erfolg des letzten Jahres in Form einer 5 1/4"-Doppelstation mit eigenem DOS, nicht erreichen können.
Mit der Herausgabe der X-Laufwerke scheinen sie es dieses Jahr allerdings doch noch zu schaffen und eine dritte Gruppe von Schneider-Usern heranzuziehen, die man am ehesten noch als "Bidirektionale" bezeichnen könnte.
Das X-System besteht aus einer 3"-Floppy mit AMSDOS-Controller sowie einem 5 1/4"-Laufwerk und dem sogenannten X-Modul. Die Zusatzfloppy ist nicht gerade eine Sensation; sie findet schon seit längerem Verwendung in der Doppelstation.
Eine Beschreibung dieses Laufwerks können wir uns schenken – die Daten sind bekannt.
Kernpunkt des Systems ist jedoch diese geheimnisvolle Pyramide mit dem Controller.
Halt – ein Controller ist das gar nicht, sondern eher eine Controller-Erweiterung!
Vortex ist bei der Entwicklung davon ausgegangen, daß es ja eigentlich Quatsch ist, dem User eine Hardware zu verkaufen, die er in Form seines AMSDOS-Controllers schon hat. Mit ein wenig Bastelei müßte es doch eigentlich hinzukriegen sein, daß mit geringem Bauteileaufwand dieser Controller ein wenig modifiziert wird.
Das X-Modul besteht aus einem kleinen Kasten mit einer Platine, die sehr gering bestückt ist. Bastler wissen, daß eine Bestückung mit wenigen Bauteilen und großer Effizienz nicht etwa ein Nachteil ist, sondern eine Meisterleistung der Hardwareentwicklung.
An der Unterseite der Platine befindet sich ein Platinenstecker, und an der Front des Kastens hängt ein Kabel mit einer Messerleiste heraus.
Diese Messerleiste wird auf den Floppyport des CPC's gesteckt und der alte AMSDOS-Controller in den Kasten auf den Platinenstecker. Danach wird noch ein Deckel auf die Unterseite des Kastens geschraubt, die Zusatzfloppy mit dem freien Stecker im DDI 1 Kabel verbunden, alles eingeschaltet und....
es passiert nichts!.
Panik in der Redaktion. Ein Blick ins Handbuch sagt uns, daß das im Controller enthaltene VDOS 2.0 ROM Nummer 5 benutzt.
Genau das gleiche ROM belegt allerdings auch unser MAXAM ROM, das ja noch hinten am Rechner steckt. Nach Entfernen des Maxam lief die Floppy auch einwandfrei, und wir bekamen endlich die ersehnte Einschaltmeldung des VDOS auf den Bildschirm.
Enttäuschungen auf Seiten der AMSDOS-Fans, die jetzt annehmen, ihre bestehende Software liefe jetzt ja immer noch nicht, sei gleich vorgebeugt. Nach Eingabe des Befehls AMSDOS vergißt das X-Modul, daß es jemals aktiviert wurde und alle Programme laufen mit dem gewohnten 3"-Format.
Beide DOS-Versionen stehen also ohne Umstecken jederzeit zur Verfügung. Software der "alten" Vortex-Doppelfloppy läuft nach wie vor. Das komfortable VDOS befindet sich abrufbereit im Speicher, und wenn man sich gar das XRS-Modul gekauft hat, hat man sogar noch eine tolle RS-232-Schnittstelle zur Verfügung. Was will man also mehr?
„Meinen Maxam wollte ich gerne wieder benutzen können, ich hab ihn so lieb", hört man den Programmierer klagen, der immer etwas zu meckern hat.
Im VDOS ist jedoch auch ein Monitor, sogar mit integriertem Disk-Monitor, enthalten. Zwar wurde im Handbuch vergessen, daß dieser Monitor im X-Modul nicht mehr mit M aufgerufen wird, sondern jetzt mit XMON gestartet wird, aber wir verraten Ihnen das an dieser Stelle. Die Eingabe von M führt nur zu einem Unknown-Command-Error.
Trotzdem ist ein Maxambegeisterter damit noch nicht zufrieden.
Schließlich hat er sich das Modul für

teures Geld gekauft und will es auch benutzen. Außerdem – vielleicht laufen dann auch andere Zusatzroms nicht mehr.
Blättern in den Anhängen der Hardware-Referenz bringt uns dann endlich auf die Lösung. Die ROM-Nummer des VDOS ist frei wählbar. Im Modul gibt es eine Steckerleiste, auf der sich eine Drahtbrücke dafür befindet. Nach Umbelegen der ROM-Nummer auf ROM 3 funktioniert auch der Maxam, und niemand hat mehr Grund zum Klagen.
Vielleicht wäre es ganz gut gewesen, diese Tatsache schon bei der Einbauanleitung zu erwähnen, man spart sich dadurch das erneute Aufschrauben des Moduls. Aber wir gehen jetzt einfach mal davon aus, daß unser Handbuch eine Vorabversion war und noch überarbeitet wird, bevor es in den Handel kommt – GELLE?
Der Zusammenbau des ganzen Systems ging flott von der Hand und sollte niemandem Schwierigkeiten bereiten. Ein falscher Zusammenbau ist nur mit aller Anstrengung möglich.
Da das Netzteil in der Floppy integriert ist, gibt es auch keinen übermäßigen Kabelsalat auf dem Schreibtisch.
Die Hardware des Systems ist also ausgeklügelt und gibt keinerlei Anlaß zur Kritik.
Die Software noch viel weniger. Hier wurde echte Präzisionsarbeit geleistet. Sowohl unter VDOS als auch unter CP/M lief alles so, wie man es erwarten kann. Neue Befehle ermöglichen das Formatieren von Basic aus, das wahlweise Booten des CP/M aus jedem Laufwerk und ein auf der Systemdiskette mitgelieferter Patch ermöglicht es sogar, das Zusatzlaufwerk ohne X-Modul als ganz normales 40-Track-Zusatzlaufwerk zu fahren.
Dieser Patch funktioniert unter Basic und CP/M sowie auf allen drei Rechnertypen.
Erfreulich ist auch die Software, die auf Diskette mitgeliefert wird. Ein komfortables Disktool und ein noch komfortableres Filecopy gehören dazu, wie die schon bekannten Utilities der Doppelstation.



Das neue Vortex-System – zwei Controller in einer Box.

Das X-System ist allerdings nicht nur als Einzellaufwerk erhältlich, sondern auch noch als Doppelstation. Laut Angaben von Vortex ändert sich an den technischen Daten nichts.
Das VDOS 2.0 ist mehr als genug beschrieben worden und benötigt keiner weiteren Erklärungen.
BUGS waren keine festzustellen, und das Ganze reiht sich nahtlos in das ansprechende Design der CPC-Reihe ein.
1 Megabyte, 2 Disk-Operating-Systems, jede Menge neue Befehle und alles noch zum sehr guten Preis von DM 758,– zeugen davon, daß Vortex noch lange nicht die Ideen ausgehen und der CPC immer erwachsener wird.
User, die bereits eine DDI 1 besitzen, können den Doppelfloppyanwendern eine lange Nase drehen. So günstig auf den letzten Stand der Technik zu kommen, war denen damals nicht beschert.
Wer schon eine Doppelfloppy hat, wird auch nicht im Stich gelassen, das X-Modul ist auch einzeln erhältlich. Ein Umbau der alten Station auf den neuen Standard ist kein Problem, und man hat noch einen VDOS-Controller übrig, aus dem man sich ein hübsches Zigarettenetui basteln kann.
Anwender, die zur Anschaffung der Doppelfloppy ihre DDI 1 verkauft haben, sollten Kleinanzeigen lesen und sich schnell wieder so ein Ding anschaffen, sonst laufen sie Gefahr, der Entwicklung hinterherzuhängen.
Auf jeden Fall lohnt sich die Anschaffung für Newcomer, die sich für kein bestimmtes Format entscheiden können.
Außerdem hat Vortex noch zwei neue Anwenderprogramme zu bieten.
Vector entspricht einer überarbeiteten Version des Programms „Topgraf" und stellt ein wissenschaftliches Grafikprogramm der Spitzenklasse dar. Das zweite Programm heißt „Maskgen" und ist ein sogenannter Job-Generator, mit dem sich problemlos Menues und Eingabemasken erstellen lassen. „Maskgen" enthält zudem noch eine RSX-Erweiterung, die das Erstellen von Jobs unterstützt.

(TM)

Logokurs
Teil 8

In diesem letzten Logokurs werden wir uns noch einmal mit der Programmierung der Grafik beschäftigen. Es werden dabei sowohl eckige, als auch sternförmige Spiralen entstehen.

Doch nun gleich zum ersten Programm:

```
?to spirale :seite :winkel :zahl
>if :zahl = 0 [stop] []
>fd :seite
>rt :winkel
>spirale (:seite + 10) :winkel (:zahl - 1)
>end
```

Dies ist ein noch relativ einfaches Programm. Probieren Sie doch einmal folgende Eingabe aus:

```
>spirale 10 90 7
```

Wie Sie vielleicht vermuten werden, ergibt die Ausführung dieses Aufrufs eine einfache, rechtwinklige Spirale. Dies deshalb, weil ein Winkel von 90 Grad angegeben wurde. Außerdem wird die Spirale sieben Seiten haben, ausgehend von dem letzten Parameter. Die erste angegebene Zahl gibt an, wie groß die erste Seite ist. Ein Abbild des Entstandenen können Sie aus Bild 1 entnehmen.

Wie Sie selbst feststellen werden, ist "Spirale" ein Programm, das zählen kann. Jedesmal wenn es sich selbst wieder aufruft (Rekursion), wird die Variable :Zahl um eins vermindert.

Natürlich läßt sich das Programm noch um ein Vielfaches erweitern – zum Beispiel so:

```
?to wechselspirale :winkel :zahl
>spirale 10 :winkel :zahl
>wechselspirale :winkel :zahl
>end
```

Erklärung:

Das Programm "Wechselspirale" benötigt zwei Angaben. Wie aus seinem Namen schon ersichtlich, handelt es sich hierbei um den Winkel und die Anzahl der Seiten einer Spirale. Nach dem Aufruf dieses Programmes wird zuerst einmal die Prozedur >Spirale< mit der festen Seitenzahl 10 aufgerufen. Anschließend wird das Ganze verdreht wiederholt.

Probieren Sie beispielsweise:

```
>Wechselspirale 90 7
>
```

Das Ergebnis dieses Aufrufs sehen Sie in Bild 2.

Doch dabei wollen wir es nicht bleiben lassen. Wir können die Größe der Wechselspiralemuster ändern, wenn

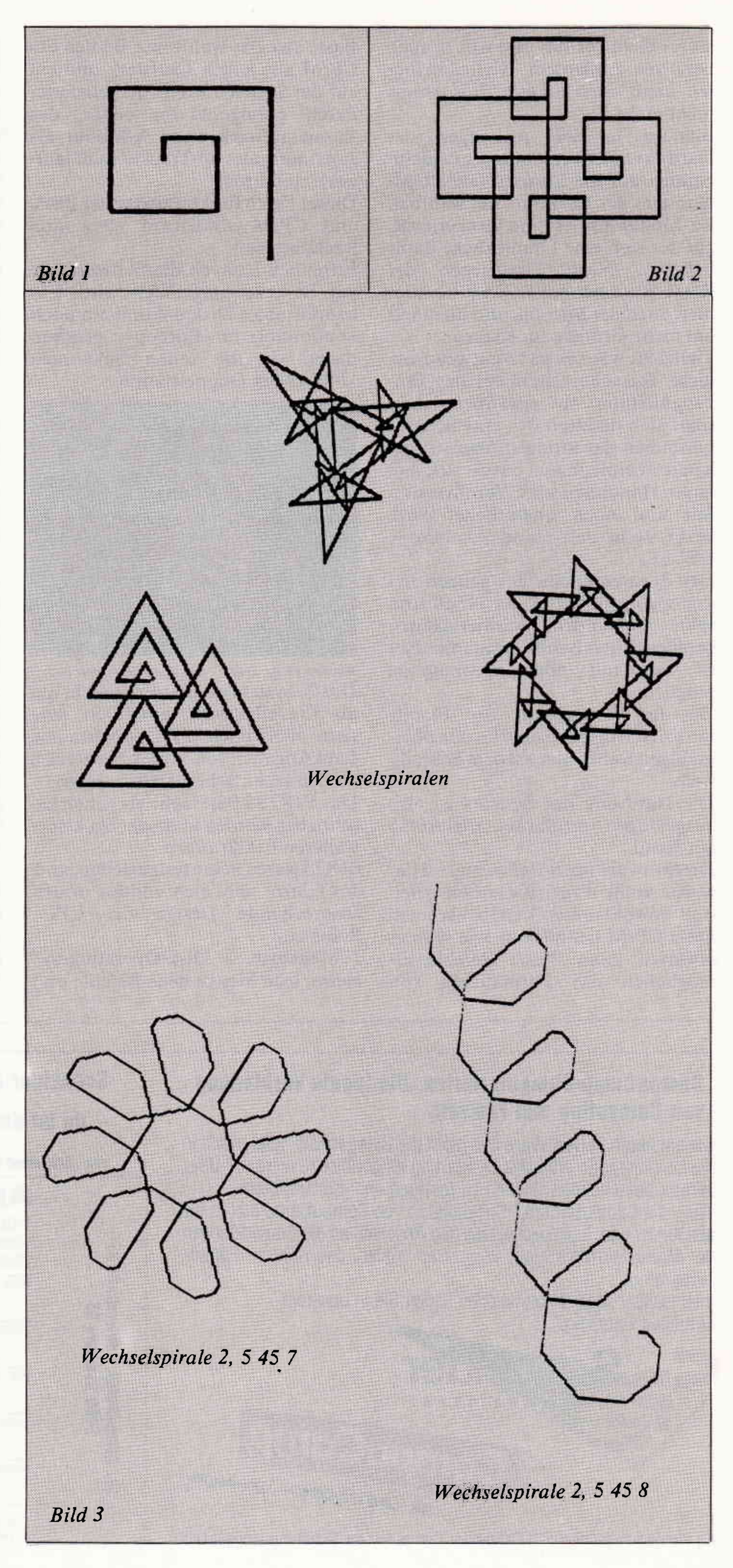


Bild 1

Bild 2

Wechselspiralen

Wechselspirale 2, 5 45 7

Wechselspirale 2, 5 45 8

Bild 3

wir den Zuwachs der Seitenlänge zu einer Unbekannten in der Prozedur "Spirale" machen und die Anfangsseitenlänge zu einer Unbekannten in >Wechselspirale<. Dies sähe dann so aus:

```
?to spirale2 :seite :winkel :zuwachs :zahl
>if :zahl = 0 [stop] []
>fd :seite
>rt :winkel
>spirale2 (:seite + :zuwachs) :winkel
  :zuwachs (:zahl - 1)
>end
```

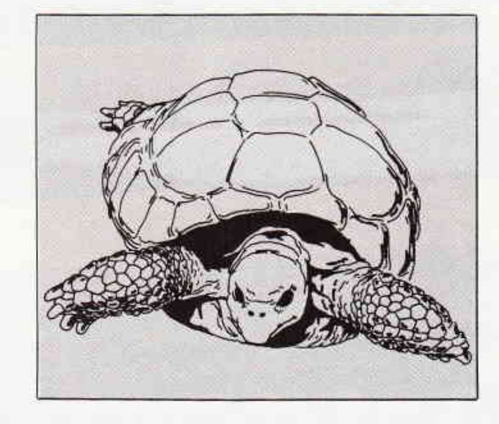
Außerdem noch Wechselspirale2:

```
?to wechselspirale2 :seite :winkel :zahl
>spirale2 :seite :winkel :seite :zahl
>wechselspirale2 :seite :winkel :zahl
>end
```

Wenn >Wechselspirale2< >Spirale2< aufruft, benutzt es denselben Wert, sowohl für die Variable :seite, als auch für die Variable :zuwachs. Mit diesem kleinen Trick bleiben die Figuren proportional zueinander.

Es entstehen mit dieser Routine sehr hübsche Figuren, wie sie in Bild 3 zu sehen sind.

Eine der interessantesten mathematischen Fragen bei den Wechselspiralen hat damit zu tun, daß einige Spiralen geschlossene Muster zeichnen, die sich ständig wiederholen, während sich



andere auf ewig um den Schirm wikkeln.

Welche Winkel- und Zahlenkombinationen zeichnen geschlossene Muster und welche offene? Um auf diese Frage eine Antwort geben zu können, muß ich ein wenig ausholen:

Es hängt damit zusammen, ob Anzahl der Winkel mal die Größe der Winkel 360, ein Vielfaches oder einen glatten Teiler davon ergeben. Doch zurück zum Kurs:

Der Schwerpunkt dieses Kurses lag beim Figurenzeichnen und der Erkundung, der dabei auftretenden Fragen. Vielleicht benutzen Sie diese Figuren auch, um Mathematik zu lernen. Sicherlich werden in nicht allzu ferner Zukunft solche Figuren und die damit zusammenhängenden Fragen fester Bestandteil der Mathematiklehrpläne sein.

Und jetzt noch ein paar kleine Tips zum Schluß:

Hoffentlich haben Sie sich von der Menge der Unbekannten, mit denen Sie es hier zu tun hatten, nicht allzusehr verwirren lassen. Falls doch, dann machen Sie es einfach so:

Der Trick liegt darin, die Wirkungsweise einer Änderung zu isolieren. Ändern Sie also jedesmal nur einen Wert. Sie werden schnell feststellen, wie die einzelnen Eingaben das spätere Ergebnis – also die Zeichnung – beeinflussen.

Außerdem ist es sehr hilfreich, vor dem Start von Dr. LOGO die Tastatur entsprechend einzustellen. Dies können Sie mit dem CP/M-Programm SETUP .COM erreichen.

Ich hoffe, Ihnen hat unser Logokurs gefallen. Möglicherweise können Sie das hier erworbene Wissen sogar erfolgreich einsetzen. Vielleicht überraschen Sie eines Tages Ihren Sohn oder Ihre Tochter damit, selbst eine der bekanntesten Computersprachen zu beherrschen.

Viel Spaß weiterhin!!!

(HF)

Panasonic KX-P 1080



Der Aufbau

Schlicht, aber solide präsentiert sich der 1080 in seinem beigen Gewand. Zum Einlegen von Endlospapier und Wechseln der Farbbandkassette lassen sich zwei Hauben auf der Oberseite des Druckers entfernen; diese Hauben sorgen während des Betriebs für die Dämmung der Druckgeräusche. Dies geschieht recht befriedigend; der Geräuschpegel hielt sich auch beim Ausdruck von Punktgrafiken in Grenzen.
Der mit neuen Nadeln arbeitende Druckkopf ist großzügig dimensioniert und wird durch seine großen Kühlrippen thermisch ausreichend entlastet. Die Druckkopfträgerplatte wird mit einer Metallbuchse auf der Führungsachse geführt und beherbergt auch die Farbbandkassette. Die gesamte Mechanik hinterläßt einen guten und soliden Eindruck. Dies macht den Drucker durchaus empfehlenswert für den Bereich der professionellen Textverarbeitung. Apropos Farbband: dies war das erste Kassettensystem in unserer Testpraxis, welches sich wirklich mit einer Hand und ohne Einfärbung der Finger wechseln ließ. Wenn schon jeder Druckerhersteller sein eigenes Kassettensystem einbaut, sollte wenigstens der Bedienungskomfort des Panasonic-Systems als Anregung für einen Standard genommen werden.
Der 1080 kann sowohl mit Einzelblättern als auch mit Endlospapier gefüttert werden. Für den Einzelblattbetrieb verfügt der Drucker über eine Friktionswalze; Endlospapier wird mittels eines Traktorantriebes weiterbefördert. Zur Führung des Endlospapiers stehen zwei mitgelieferte Drahtbügel zur Verfügung. Leider ist keine Abrißkante zur Trennung zweier Papierbogen vorhanden; dies sollte in dieser Preis-/Leistungsklasse eigentlich obligatorisch sein, weil diese Kante das Handling mit Endlospapier doch wesentlich komfortabler macht.
An der Oberseite des Druckers befinden sich zwei Bedienungsfelder zur Anwahl einiger Funktionen; die Tasten sind erfreulicherweise nicht in der üblichen Folienbauweise ausgelegt. Neben den Zeilen/Formularvorschub-Funktionen, die auch im Online-Betrieb arbeiten, ist die Festeinstellung der Standard-, NLQ- und Schmalschrift per Schalter möglich.

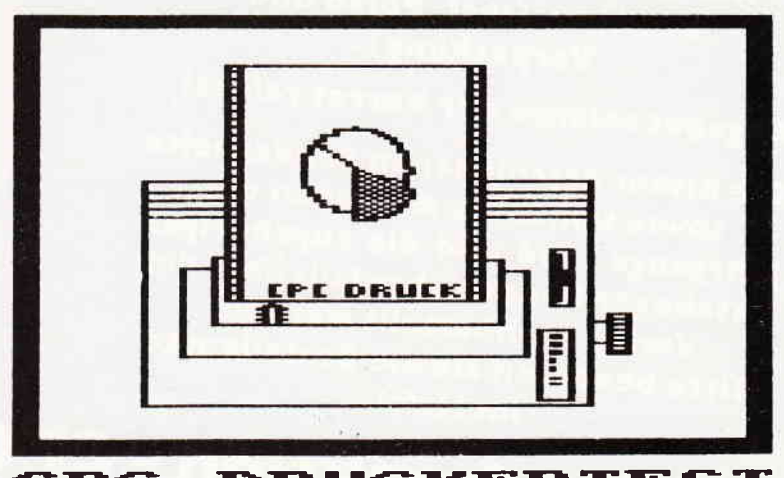
Weitere Dauerfunktionen wie z.B. die Anwahl eines landesspezifischen Zeichensatzes oder Unterdrückung des Zeilenvorschubes nach jeder Zeile lassen sich über einen achtpoligen DIP-Schalter einstellen, der allerdings erst nach Entfernen der Schalldämmhauben recht umständlich im Inneren des Druckers zu erreichen ist.

Die Hardware

Der Panasonic wird serienmäßig mit einer Centronics-Schnittstelle ausgeliefert, die die Byteweise (parallele) Übertragung der Daten zum Drucker ermöglicht und problemlos mit den CPC-Rechnern zusammenarbeitet. Optional wird auch eine serielle Schnittstelle (RS 232C) angeboten; diese ist jedoch für die Schneider-User weniger interessant.

Die Druckmatrix beträgt beim Ausdruck der Standardschriften 9 x 9 Punkte; NLQ wird in einer 18 x 18-Matrix hergestellt. Dem durch die Ansteuerbarkeit der Epson-Drucker gesetzten Standard folgend, ist der 1080 in der Lage, Punktgrafiken mit einer Dichte von bis zu 1920 Punkten pro Zeile darzustellen. Der Ausdruck

... und eine Hardcopy ...



```
     Standardzeichensatz
 !"#$%&'()*+,-./0123456789:;<=>?§ABCDEFGHIJKLM
PQRSTUVWXYZÄÖÜ^_'abcdefghijklmnopqrstuvwxyzäöüß

     Sonderzeichen
[\]{|}~°ç§éùè¨ÄÖÜäöüß[\]{|}~
ÆØÅæøå~ÄÖÅäöåü°\éàòèì¡Ñ¿¨ñ}~
```

von Standard- und NLQ-Text erfolgt bidirektional und druckwegoptimiert; Grafik wird einseitig von links nach rechts gedruckt, wobei auch keine Optimierung stattfindet.
Wie in dieser Preisklasse für Matrixdrucker üblich, können neben einem Original auch zwei Kopien erzeugt werden.

Die Steuerung

Wie erwähnt, ist der Panasonic-Drucker voll kompatibel zu den Steuercodes im Epson-Standard, allerdings ist es uns nicht gelungen, im NLQ-Modus hoch- oder tiefgestellte Zeichen auszudrucken (siehe Standardausdruck).
Im Drucker-ROM stehen die Sonderzeichen für elf landesspezifische Zeichensätze zur Verfügung, die sowohl durch Steuercodes als auch durch die eingangs erwähnten DIP-Schalter aktiviert werden können. Die Möglichkeiten der Schriftform sind aus dem Standardausdruck ersichtlich.
Als Sonderfunktionen sind der Selbsttest und der Hexadezimaldruck zu nennen, welche beim Einschalten des Druckers per Tastendruck aufgerufen werden können. Die Hexdruck-Funktion ermöglicht den Ausdruck der ASCII-Werte aller vom Rechner gesendeten Daten (außer Steuerzeichen) in hexadezimaler Form.

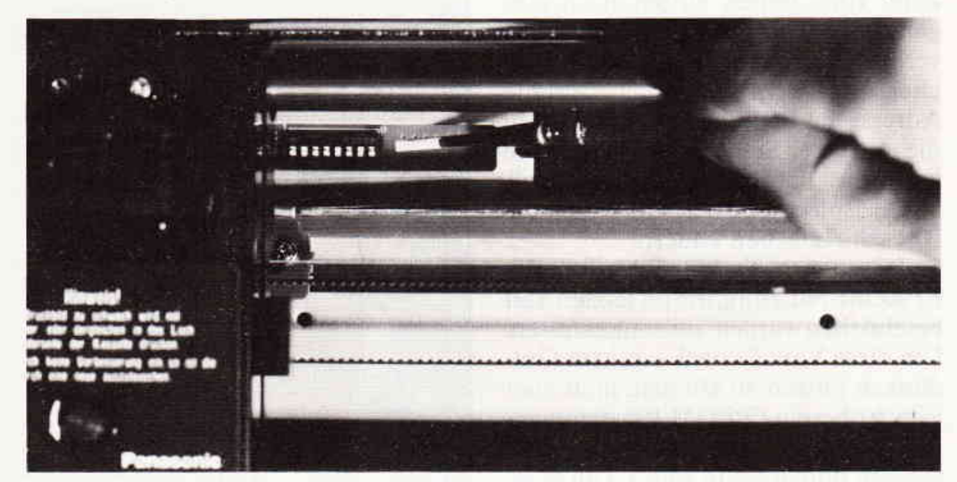
Zusammenfassung

Der KX-P 1080 hinterläßt durch die solide Bauweise und seine universelle Anwendbarkeit einen rundum guten Eindruck, ein Einsatz als Arbeitspferd ist durchaus denkbar. Die fehlende Papierabrißkante schmälert zwar nicht die Betriebsleistungen dieses Druckers, jedoch sollte man dieses Utensil zum Standard bei Geräten dieser Preisklasse machen.
Dennoch ist der Panasonic auch für gehobene Leistungsansprüche empfehlenswert; der Käufer erhält eine angemessene Leistung für den Kaufpreis.

Technische Daten

Schnittstellen:
Centronics serienmäßig, RS 232 C optional
Druckarten:
Standard/NLQ/Grafik
Zeichen/Zeile:
Pica 80, breit 40, Elite 96, breit 48, condensed 137
Geschwindigkeit:
Standard 100 Z/s, condensed 86 Z/s, NLQ 20 Z/s
Druckrichtung:
bidirektional und druckwegoptimiert
Papier:
Einzelblatt/Endlospapier
1 Original, 2 Kopien
Druckkopf:
9-pin, Lebensdauer ca. 100 Mio. Zeichen
Preis:
ca. DM 1140,–
Info:
Panasonic Deutschland GmbH
2000 Hamburg 54

(ME)




```
     Schriftarten
Dies ist Text in Near-Letter-Quality , enlarged  und condensed
Dies ist Text im Elite - Schriftmodus , enlarged  und condensed
Dies ist Text im Pica - Schriftmodus , enlarged  und condensed
Dies ist NLQ im Italic-Modus, enlarged  und condensed
Dies ist Elite im Italic-Modus, enlarged  und condensed
Dies ist Pica im Italic-Modus, enlarged  und condensed
Dies ist der FETTSCHRIFT - Modus, enlarged  und condensed
Dies ist der Doppelanschlag-Modus, enlarged  und condensed


      Weitere Moeglichkeiten
NLQ  :Text hoch-und tiefgestellt,dabei unterstrichen
Elite :Text hoch-und tiefgestellt,dabei unterstrichen
Pica  :Text hoch-und tiefgestellt,dabei unterstrichen
```

EPROMs am CPC 464

Teil 2

Der erste Teil unseres Berichtes in Heft 5/86 behandelte ausführlich das Thema der Nutzung von mit eigenen Programmen gefütterten Eproms an den Schneider Computern mittels einer Eprom-Karte.
In diesem Beitrag wollen wir uns ebenso ausführlich damit befassen, wie die Bits, die wohlgeordnet unser späteres Programm darstellen, in die Speicherbausteine eingebrannt werden.

EPROMs werden durch das Betriebssystem des CPC optimal unterstützt. Beachtet man einige Konventionen, so werden darin enthaltene RSX-Befehle beim Einschalten automatisch eingebunden. Die RSX-Routinen selbst verbrauchen dabei keinen RAM-Speicher. Außerdem wird eine bestimmte Adresse des EPROMs angesprungen; die an dieser Stelle liegende Routine kann das Betriebssystem "patchen", also beispielsweise den Mode oder die Bildschirmfarben ändern.
So lassen sich die Möglichkeiten der EPROM-Nutzung, die im letzten Teil beschrieben wurden, zusammenfassen. Um diese Vorteile nach eigenem Gutdünken nutzen zu können, muß man jedoch über ein EPROM-Programmiergerät verfügen. Die Schaltung eines solchen finden Sie in Bild 1. Um es jedoch gleich vorwegzunehmen:
Die hier beschriebene Schaltung ist – ihrem Preis entsprechend – dazu gedacht, einzelne EPROMs zu "brennen". Sollten Sie beabsichtigen, häufiger von ihnen Gebrauch zu machen, dann empfiehlt sich der Kauf eines fertigen EPROMMERs. Ein solches Gerät (mit Software) ist im (Versand-) Handel für ca. 250,– DM erhältlich und bietet erheblich größeren Komfort als die hier vorgestellte Schaltung.
Der wesentlichste Unterschied wird wohl in der "Write Only"-Eigenschaft unserer Schaltung bestehen; das heißt, daß Sie mit dieser Schaltung ein EPROM eben nur brennen können. Zum Lesen des EPROM-Inhaltes, also auch zum Verifizieren und zur Nutzung, benöti-

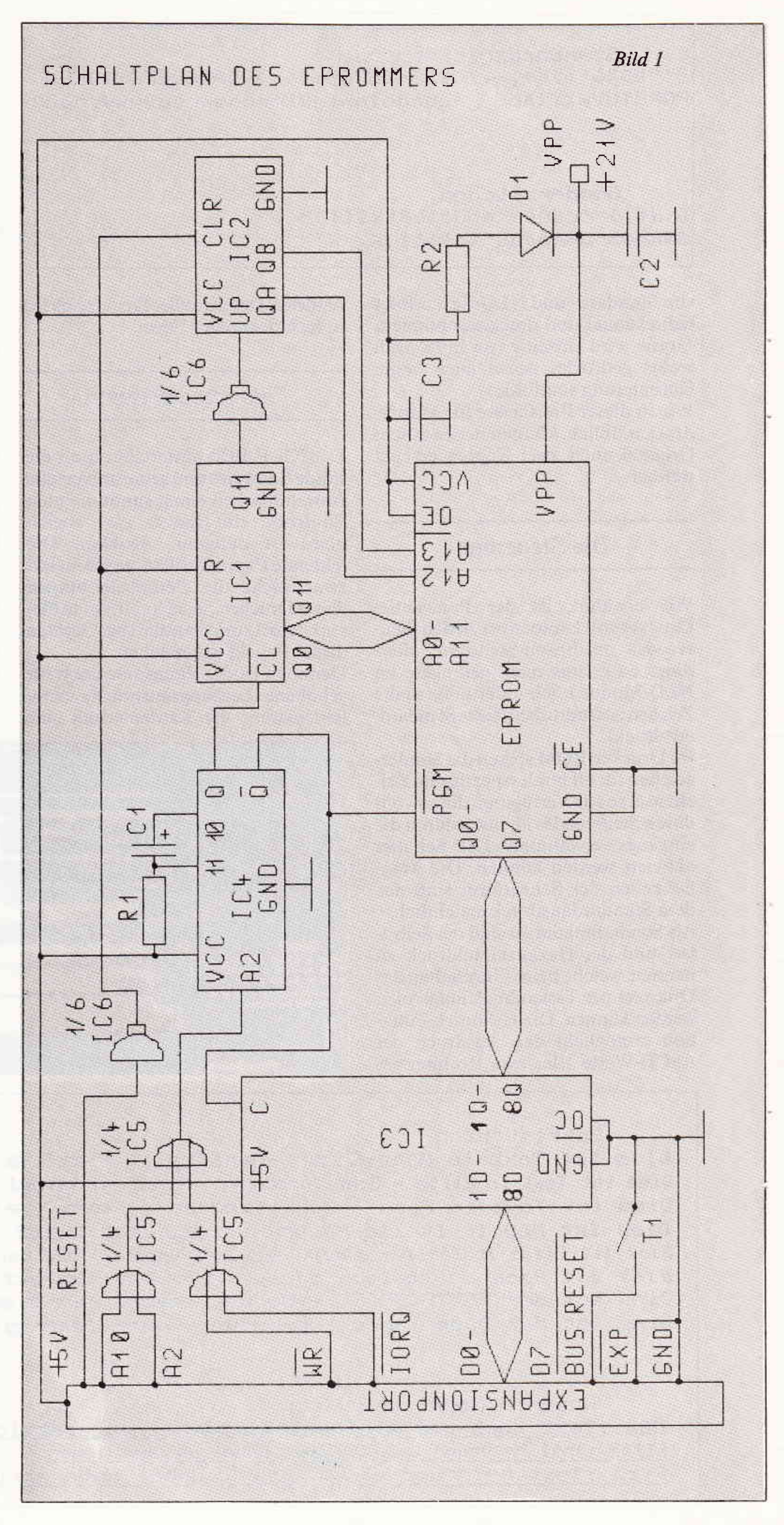


Bild 1

gen Sie die im letzten Heft vorgestellte Schaltung.
Bevor ich jedoch auf die Schaltung näher eingehe, einige...

Grundlagen

Die Speicherzelle eines EPROM besteht aus einem MOS-FET, also einem Metal-Oxyde-Semiconductor-Field-Effect-Transistor. Ein solcher Transistor hat die Eigenschaft, wie der Name schon sagt, mit elektrischen Feldern zu arbeiten. Stark vereinfacht kann man aber Folgendes sagen:
Wenn man die Pole eines Gleichspannungsfeldes (z.B. die Platten eines Kondensators) ideal isoliert, dann fließt kein Strom, das heißt, die Ladungsträger verharren bis auf weiteres auf den Polen.
In einem EPROM sind aber die Steuerelektroden (Gates) der FETs völlig von isolierendem Siliziumoxid umgeben. Eine einmal auf ein Gate aufgebrachte Ladung bleibt dort also für etwa 10 Jahre bestehen. Die Höhe der Spannung am Gate steuert aber wiederum den Strom durch die Drain-Source-Strecke (Drain und Source sind die beiden anderen Anschlüsse eines FET: Zufluß und Quelle), welcher dann als Maß für die auf dem Gate gespeicherte Information "Ladungsträger ja oder nein" gelten kann; und schon haben wir ein "Bit".
Um Ladungsträger auf ein Gate zu bringen, müssen sie soviel Energie besitzen, daß sie in der Lage sind, die Siliziumoxidbarriere zu überwinden.
Diese Energie erhalten sie durch die Spannung Vpp, die während der Programmierung 21 V beträgt.
Die jetzt auf dem Gate befindlichen Ladungsträger sehen vorläufig keine Veranlassung mehr, das Gate zu verlassen, denn sie haben nicht mehr genug Energie, das Siliziumoxid noch einmal zu überwinden. Die Bestrahlung mit energiereichem UV-Licht ionisiert jedoch das isolierende Siliziumoxid, so daß die Ladungsträger in der Lage sind, trotz ihrer geringen Energie die Barriere wieder zu überwinden und abzufließen; das EPROM wird gelöscht.

Hardware

Das Blockschaltbild der Schaltung, die am 50-poligen Systembus betrieben wird, sehen Sie in Bild 2. Bei genauerer Betrachtung kann man ihm einiges über die Funktionsweise unseres EPROMMERs entnehmen.

Die Hauptteile der Schaltung sind:
- Adressgenerator CD4040 & 74LS193
- Datenzwischenspeicher 74LS373
- Impulsgenerator 74121
- Dekodierung 74LS32

Nach dem Einschalten des Rechners (die Schaltung wird von ihm mitversorgt) wird der Adressgenerator zurückgesetzt; alle Adressleitungen A0 bis A13 sind low.
Bei einem Programmiervorgang geschieht folgendes:
Die gesamte Schaltung wird mittels des BASIC-Befehls "Out &FBFB,wert" angesprochen. Die Dekodiergatter um IC 5 erkennen diesen Out-Befehl an den CPC-Systembusausgängen WR (Ausgabesignal des Microprozessors), IORQ (Ausgabe betrifft IO = Ein-/Ausgabebausteine) und A2/A10 (Microprozessoradressausgänge). Der Ausgang der Gatterkombination aktiviert (triggert) das Monoflop 74121.
Das Monoflop 74121 ist ein Baustein, der einen Impuls erzeugt, dessen Länge von einem RC-Glied (R1/C1) abhängt. Dieser Impuls steht an den Ausgängen Q und Q' (sprich Qquer) in gegenphasiger Form zur Verfügung; das heißt Q ist low wenn Q' high ist, und umgekehrt. Die Impulslänge T folgt der Formel: $T = 0{,}7 * R * C$.
Die Impulslänge in unserer Schaltung ist 50 ms, also gleich der vom EPROM-Hersteller vorgeschriebenen Länge des Programmierimpulses.
Der Ausgang Q triggert den Takteingang des CD4040, also des Adressgenerators, und erhöht so die aktuelle Adresse um 1.
Der Ausgang Q' aktiviert den Eingang LE des 74LS373 und sorgt so dafür, daß die zum Zeitpunkt der abfallenden Flanke des Impulses an den Eingängen des 74LS373 anliegenden Daten übernommen und an den Ausgängen angelegt werden. Die Ausgänge wiederum führen an die Eingänge des EPROM und stellen dort die einzubrennenden Daten statisch zur Verfügung.
Außerdem führt Q' zum Anschluß PGM des EPROM und dient hier als Programmierimpuls.

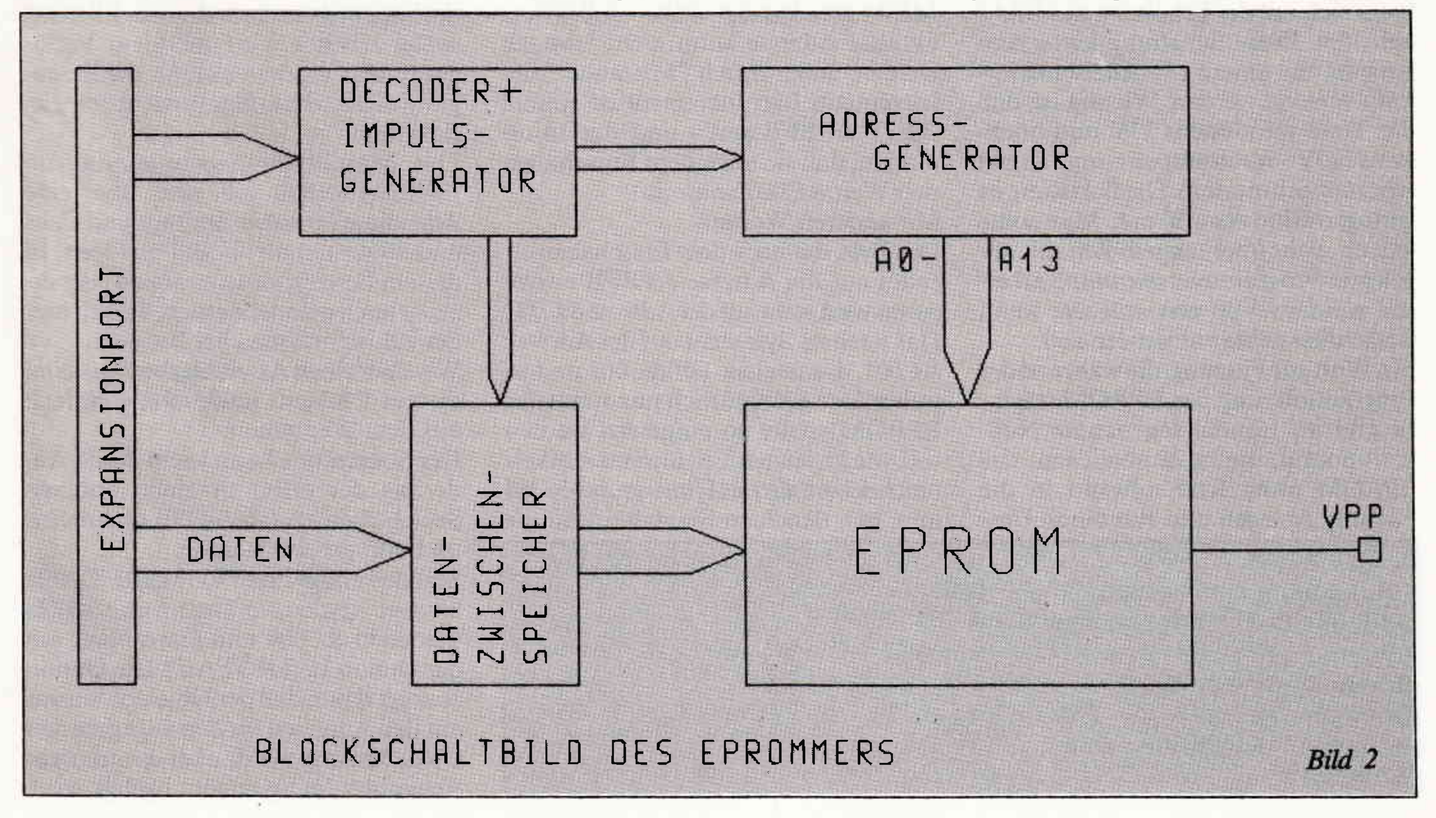


Bild 2

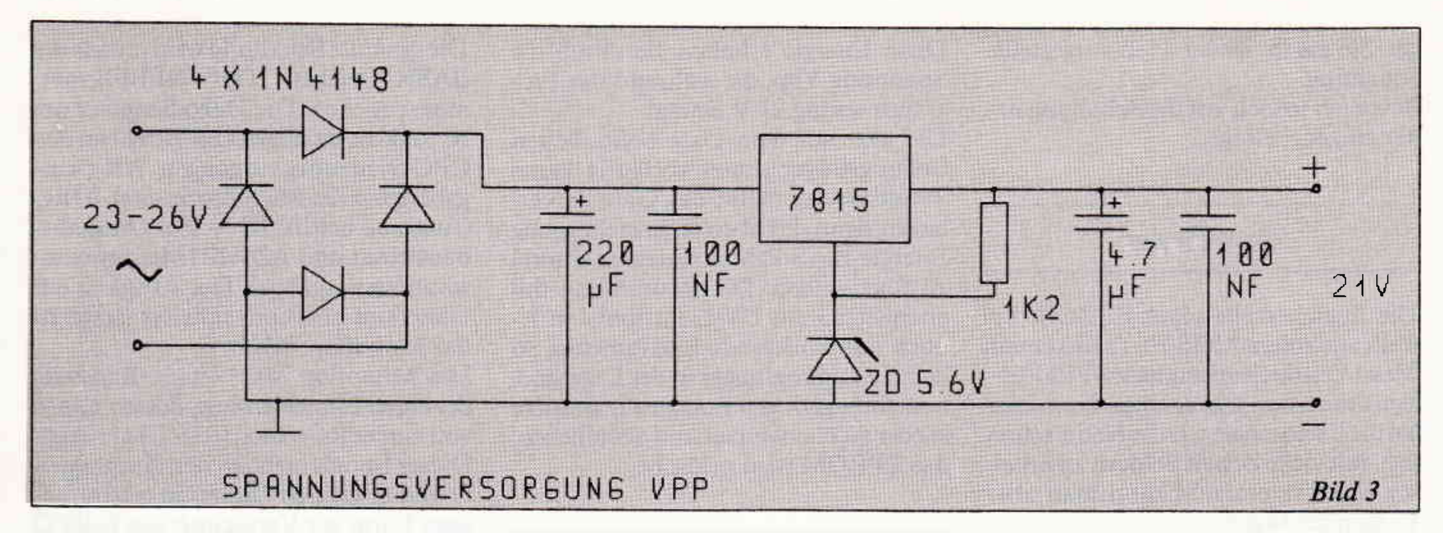


Bild 3

Das alles noch einmal in Kurzfassung:
- die Schaltung wird mittels "OUT &FBFB,wert" angesprochen.

Dieses Ansprechen erzeugt automatisch:
- den 50ms-Programmierimpuls
- einen Increment des Adressgenerators
- das Zwischenspeichern der Daten

Schwierigkeiten bereitet die Programmierspannung Vpp, die extern zugeführt werden muß. Diese Spannung sollte erst nach dem Einschalten des Rechners angelegt werden; der dabei eventuell entstehende Funken (Ladung von C2) kann jedoch die Schaltung beeinträchtigen, zum Beispiel ein Bit des bereits zurückgesetzten Adressgenerators wieder setzen. Deshalb sollte man in folgender Reihenfolge vorgehen: Zuerst den Rechner einschalten, dann Vpp anlegen und jetzt den Rechner nochmals mittels Taste 1 zurücksetzen. Wer kein stabilisiertes Netzteil zur Erzeugung der Vpp zur Verfügung hat, kann sich mit der Schaltung aus Bild 3 behelfen. Diese Schaltung kann zum Beispiel mit einem Modelleisenbahntrafo versorgt werden. Wichtig ist, daß der Trafo mindestens 23V und höchstens 25V Wechselspannung abgibt (vorher nachmessen). Die Belastung ist gering: es sind etwa 50 mA. Man sollte jedoch nicht über längere Zeit mit angelegter Programmierspannung arbeiten, sondern Vpp erst anlegen, wenn tatsächlich gebrannt werden soll.
Ein Wort zur Fassung, die während der Programmierung das EPROM trägt: Es gibt im Handel sogenannte Nullkraft-Sockel, die es ermöglichen, das EPROM ohne Kraftaufwand in die Fassung zu legen und erst durch Umlegen eines kleinen Hebels den Kontakt herzustellen. Diese Fassungen sind vergleichsweise sehr teuer; trotzdem lohnt sich der Mehraufwand, sobald ein wenig mit EPROMs experimentiert werden soll. Lediglich diejenigen, die nicht mehr und nicht weniger planen, als die Software einzubrennen, die im nächsten Heft vorgestellt wird, können sich diese Ausgabe sparen. In diesem Zusammenhang lohnt sich auch der Vergleich von Einzelhandel und Versand: Der Preisunterschied beträgt bis zu 100 %.

Software

Wie oben bereits gesagt, wird der EPROMMER mit "OUT &FBFB,wert" von BASIC aus angesprochen. Die Variable "wert" steht hier für dasjenige Byte, das in der aktuellen Adresse, auf die der Adressgenerator des EPROMMERs gerade zeigt, stehen soll. Diese aktuelle Adresse kann nicht ermittelt werden, außer durch "Mitzählen" der Increments (ein Increment ist einmal "OUT &FBFB,wert") und der Information, daß sie nach dem Einschalten den Wert &C000 annimmt.
Mit anderen Worten:
Das Byte, das nach dem Einschalten als erstes auf die Adresse &FBFB ausgegeben wird, steht auf der Adresse &C000. Das nächste Byte steht auf der Adresse &C001, das nächste auf &C002 und so weiter. Das gilt natürlich nur, wenn das EPROM später so eingesetzt werden soll, wie im letzten Teil unseres Artikels beschrieben; also als Hintergrund-ROM im CPC. Beachten Sie dabei, daß die Bytes immer nur auf eine (!) Adresse (&FBFB) ausgegeben werden, aber später hintereinander im Speicher stehen.
Das Problem besteht darin, daß die Daten- oder Programmbytes, die in das EPROM sollen, nicht aus dem Speicherbereich kommen können, in dem sie später stehen werden, da ab &C000 der Bildschirmspeicher liegt.
Also muß ein weiterer Speicherbereich definiert werden, nämlich der, in dem das Programm auf seine Ausgabe wartet. Außerdem müssen wir eine Variable einführen, die eine Adresse beinhaltet, die auf das aktuelle auszugebende Byte innerhalb dieses Speicherbereiches zeigt.
Das Programm aus Listing 1 bewältigt diese Aufgaben und geht davon aus, daß die EPROM-Daten in einem Binärfile namens ROMSOFT auf Kassette vorliegen.
Aus der schon erwähnten Tatsache, daß der Speicherbereich, in dem das EPROM später arbeiten wird, nicht zur Verfügung steht, ergeben sich für die Erzeugung dieses Binärfiles gewisse Schwierigkeiten.
Das Maschinenprogramm, das im EPROM stehen soll, muß über eine Adresslage (absolute Sprünge und Calls und so weiter) ab &C000 verfügen. In diesem Bereich kann es aber weder erzeugt noch getestet werden, da sich hier der Bildschirmspeicher befindet.
Wer über einen Assembler verfügt, kann diesem Problem noch relativ einfach aus dem Weg gehen:
Der Sourcecode kann leicht durch Änderung des ORG in einen anderen Speicherbereich (ab &4000) assembliert und dort getestet werden.
Soll das endgültige "ROMSOFT"-File erzeugt werden, so ändert man ORG wieder in &C000 und assembliert mit der Option 16 (DEVPAC). Die Option 16 sorgt dafür, daß der Objektcode zwar mit den Adressen für den angegebenen ORG versorgt wird, aber hinter dem Assembler im Speicher abgelegt wird.

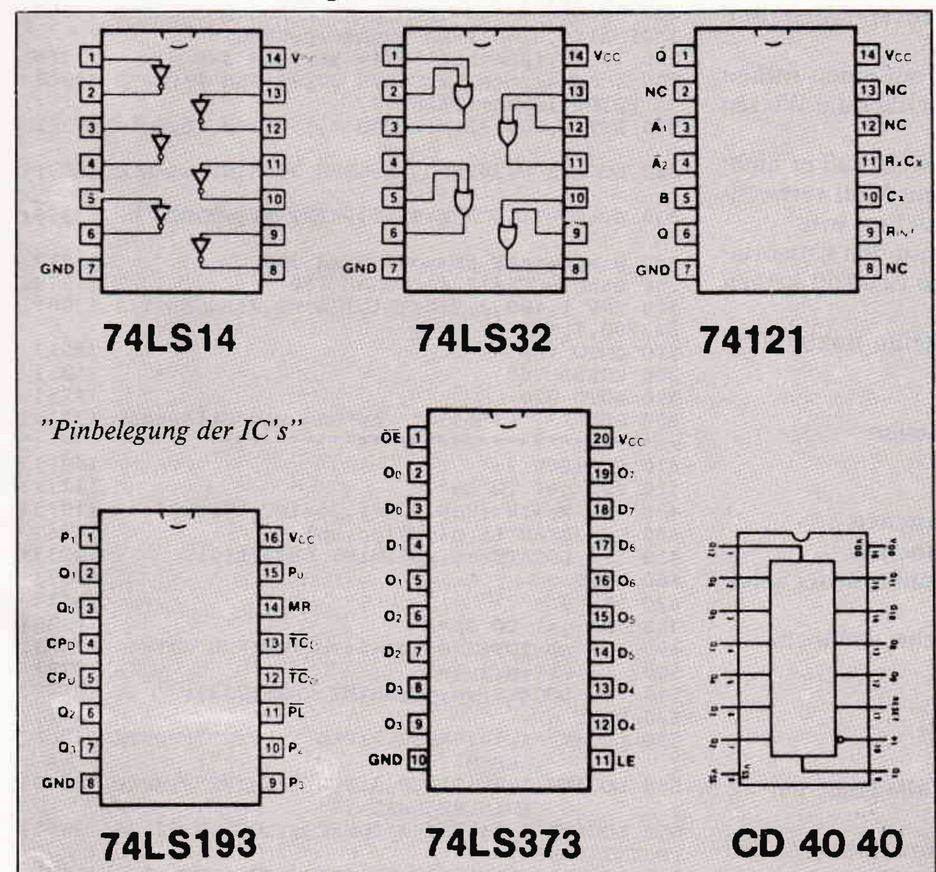
Listing 1

```
10 MEMORY &3FFF
20 LOAD"romsoft.bin",&4000
30 FOR wertadresse=&4000 TO &7FFF
40 wert=PEEK(wertadresse):OUT &FBFB,wert:PRINT HEX
$(wertadresse),
50 FOR zeit=1 TO 50:NEXT zeit
60 NEXT wertadresse
70 END
```

Dort ist er natürlich auf Grund der "falschen" Adresslage nicht lauffähig. Mit 'LOAD"ROMSOFT",&4000' kann das File dann in den Speicher ab &4000 geladen werden.
Diejenigen, die nur ein Monitorprogramm (wie in anderen Ausgaben von CPC International bereits vorgestellt) besitzen und damit ein Maschinenprogramm "zu Fuß" erstellen wollen, sehen harten Zeiten entgegen. Sie sollten das "ROMSOFT"-File gleich für die endgültige Adresslage ab &C000 schreiben und dabei so gewissenhaft wie möglich vorgehen, um eine allzu große Anzahl Fehler zu vermeiden. Einige Übung im Umgang mit OP-Codes gehört aber auf jeden Fall dazu. Für die Änderung lediglich einiger Voreinstellungen (z.B. Mode) oder die Einführungen weniger kurzer RSX-Befehle mag aber auch diese Methode gangbar sein.
Diejenigen, die weder über einen Assembler noch über einen Monitor verfügen, verweise ich aber nochmals auf die nächste Ausgabe von CPC International, in der ein menuegesteuertes "Patchprogramm" und einige schlüsselfertige RSX-Befehle vorgestellt werden. Nach Durchlaufen des Patchers steht dann ein fertiges "ROMSOFT"-File bereit.
Nun aber zurück zu unserem Programm: In Zeile 10 wird Himem auf &3FFF gesetzt, um mit 'LOAD"ROMSOFT", 64000' in Zeile 20 das Binärfile laden zu können.
Zeile 30 eröffnet die Ausgabe "For-Next"-Schleife. Zeile 40 beinhaltet den bekannten "OUT &FBFB,wert" sowie einen Print-Befehl, der das Fortschreiten der Programmierung anzeigt. Die Schleife in Zeile 50 dient der Verzögerung, um dem EPROMMER Zeit zu geben, den 50 ms-Impuls zu beenden. In Zeile 60 wird die Ausgabeschleife beendet und mit ihr das Programm (Zeile 70).
Wichtig ist, daß ein eventuell unbenutzter Teil des EPROM entweder mit FFH bzw. 255d gefüllt oder die Programmierung bei der Endadresse des EPROM-Programmes abgebrochen wird. Dadurch behält man es sich vor, nachträgliche Erweiterungen einzubrennen, ohne vorher das EPROM löschen zu müssen.

Listing 2

```
10            ORG   #C000
20            DEFB  #01
30            DEFB  0,0,0
40            DEFW  #C00F
50            JP    INIT
60            JP    HOCH
70            JP    RUNTER
80            DEFM  "INI"
90            DEFB  "T"+#80
100           DEFM  "HOC"
110           DEFB  "H"+#80
120           DEFM  "RUNTE"
130           DEFB  "R"+#80
140           DEFB  0
150 INIT:     LD    A,2
160           CALL  #BC0E
170           LD    HL,MSGS
180 LOOP:     LD    A,(HL)
190           OR    A
200           JR    Z,FIN
210           CALL  #BB5A
220           INC   HL
230           JR    LOOP
240 FIN:      SCF
250           RET
260 MSGS:     DEFM  "CPC international EPROM-Software"
270           DEFB  0
280 HOCH:     LD    B,1
290           CALL  #BC4D
300           RET
310 RUNTER:   LD    B,0
320           CALL  #BC4D
330           RET
```



"Pinbelegung der IC's"

Die Stückliste

EPROM 2764 oder 27128
IC 1 CD 4040
IC 2 74LS193
IC 3 74LS373
IC 4 74121
IC 5 74LS32
IC 6 74LS14
C1 ELKO 4,7 microFarad
C2 0,1 microFarad
C3 Kerko 0,1 microF (evtl. mehrere)
R1 15 kohm
T1 Taste 1xein
Steckverbinder (Slot 2x25 pole)
Fassungen (Lochrasterplatine)

(J. Hüpper/ME)

PERGO



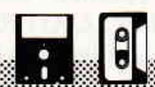
PERGO arbeitet als Parkwächter in den städtischen Anlagen von Lünchen. Seine Aufgabe ist es, am Ende eines jeden Tages auf den Wiesen des Parks für Ordnung zu sorgen und den ganzen Müll, den Besucher achtlos haben fallen lassen, wieder einzusammeln.
Dummerweise ist der Stadtpark nicht nur Ziel harmloser Ausflügler, sondern auch noch Pausenhof der nahegelegenen Terroristenschule. Diese Jungchaoten benutzen die Wiesen als Übungsgelände für Minenvergraben und Bombenlegen.
Pergo rennt nun also in der Hoffnung auf einen frühen Feierabend mit seinem Minensuchgerät über die Wiesen und räumt auf.
Dabei muß er natürlich höllisch aufpassen, daß er nicht versehentlich die versteckten Bomben mit Müll verwechselt und nicht auf die Spur des Minensuchers tritt.
Steuern Sie den fleißigen Parkwächter mit den Cursortasten und achten Sie darauf, daß ihm ein Fehltritt keinen vorzeitigen Dienstschluß bereitet.

(Stefan Bokler/TM)

Variablen:

QQ,Q	Zähler bei For-Next-Schleifen
LEBEN	Anzahl der Leben
SP	Spielstärke
X	X-Position von dem Männchen
Y	Y-Position von dem Männchen
PZ	Anzahl der gefressenen Punkte in der Spielstärke
RI	Richtung (1=links, 2=rechts, 3=hoch, 4=runter)
RA	alte Richtung
STRICH$	Art der Linie (┌;┐;└;┘;-;|)
QQ$	Abfrage der Tastatur
BI (24,41)	Inhalt des Bildschirmes (alle gesetzten Punkte)
PP	zur Überprüfung, ob Bonusspiel (bei den Mehrfachen von 1500)

für 464-664-6128



```
10 REM **** Coproduktion Stefan Bokler & B [5450]
oris Roth ***************************
20 REM **** Version 5.04 ********* Maerz 1 [3264]
986        ***************************
30 REM **** Pergo                          [1296]
           ***************************
40 REM ************ ZEICHENERKLAERUNG **** [3909]
***********************************
50 INK 0,0:INK 2,9:INK 3,6:INK 1,22 ' setz [1780]
t Farbscala
60 ON ERROR GOTO 950                       [1551]
70 MODE 1:CLS:BORDER 3:PAPER 0:PEN 1:LOCAT [4404]
E 1,2:DIM bi(24,41)
80 PRINT "Zeichen    Punkte":PEN 2:PRINT   [3319]
90 FOR Q=1 TO 4                            [951]
100  :   READ FI,PU,P4                     [874]
110  :   PRINT "  ";CHR$(FI);"        ";p [1953]
u
120 NEXT Q                                 [367]
130 PEN 1:LOCATE 5,13:PRINT" Coproduktion  [4406]
: Stefan & Boris"
140 LOCATE 17,15:PRINT"PERGO"              [1336]
150 MOVE 50,150:DRAWR 530,0:DRAWR 0,72:DRA [3875]
WR -530,0:DRAWR 0,-72
160 leben=3:ENT 1,100,-2,2:SOUND 1,142,140 [2657]
,15,0,1
170 REM *********** Hauptprogramm ******** [3963]
*************************************
180 SP=sp+1:x=20:y=11:PZ=0:RI=0:ra=0:stric [4466]
h$=CHR$(249)
190 FOR Q=1 TO 40:FOR QQ=1 TO 23:BI(QQ,Q)= [3555]
0:NEXT QQ:NEXT Q
200 FOR q=1 TO 15:qq$=INKEY$:NEXT          [1409]
210 GOSUB 400                              [964]
220 GOSUB 600                              [925]
230 IF x<1 OR x>40 OR y<1 OR y>23 OR bi(y, [3110]
x)=5 THEN 850 ' Kontrolle
240 IF bi(y,x)<1 THEN 380                  [592]
250 PRINT CHR$(7);                         [1175]
260 pp=INT(punkte/1500)                    [1197]
270 IF bi(y,x)=1 THEN punkte=punkte+50 ELS [2631]
E IF bi(y,x)=2 THEN punkte=punkte+25
280 IF bi(y,x)=3 THEN punkte=punkte+10 ELS [4180]
E IF bi(y,x)=4 THEN punkte=punkte+5
290 IF INT(punkte/1500)= pp THEN 320       [1468]
300 leben=leben+1:LOCATE 32,24:PEN 1:PAPER [4382]
 3:PRINT leben;:PAPER 0
310 ENT 1,100,-2,2:SOUND 1,142,140,15,0,1  [1262]

320 LOCATE 9,24:PEN 1:PAPER 3:PRINT punkte [2591]
;
330 ST=ST-1:LOCATE 9,25:PRINT ST;:PAPER 0  [3360]

340 pz=pz+1:IF pz<sp*7 THEN 380            [1912]
350 mann$=CHR$(224):GOSUB 760              [1378]
360 ENV 1,100,2,2:ENT 1,100,-2,2:SOUND 1,2 [2981]
84,200,1,1,1
370 GOTO 180                               [401]
380 GOSUB 760                              [925]
390 GOTO 220                               [425]
400 REM ************** Aufbau vom Bildschir [3319]
m ***********************************
410 CLS:PEN 2                              [491]
420 FOR q=1 TO sp*10                       [1271]
430  :   xp=INT(RND*40)+1:yp=INT(RND*23)+1  [1913]
440  :   READ fi,p4,bi(yp,xp)              [656]
450  :   LOCATE xp,yp:PRINT CHR$(fi);      [2233]
460 NEXT q                                 [367]
470 PEN 3                                  [547]
480 FOR q=1 TO sp*2                        [1238]
490  :   xp=INT(RND*40)+1:yp=INT(RND*23)+1  [1913]
500  :   bi(yp,xp)=5                       [972]
510  :   LOCATE xp,yp:PRINT CHR$(252);     [2127]
520 NEXT q                                 [367]
530 LOCATE 1,24:PAPER 3:PEN 1:PRINT"PUNKTE [4061]
:          LEBEN         :          ";
540 LOCATE 1,25:PAPER 3:PEN 1:PRINT"STERNE [5379]
:          SPIELSTAERKE :          ";
550 LOCATE 9,24:PRINT PUNKTE;:LOCATE 32,24 [3451]
:PRINT LEBEN;
560 ST=SP*7:LOCATE 9,25:PRINT ST;:LOCATE 3 [2703]
```

```
2,25:PRINT SP;
570 PAPER 0                                    [816]
580 LOCATE x,y:PRINT" ":bi(y,x)=0              [1468]
590 RETURN '   Aufbau beendet                  [1624]
600 REM ************* Richtung  bestimmen      [3949]
**********************************
610 ra=ri                                      [167]
620 IF JOY(0)=1 AND ri<>2 THEN ri=1 ' prue     [3000]
ft Joystik
630 IF JOY(0)=2 AND ri<>1 THEN ri=2            [1307]
640 IF JOY(0)=4 AND ri<>4 THEN ri=3            [2392]
650 IF JOY(0)=8 AND ri<>3 THEN ri=4            [1693]
660 ta$=INKEY$                                 [628]
670 IF ta$=CHR$(242) AND ri<>4 THEN ri=3 '     [3375]
 prueft Cursor
680 IF ta$=CHR$(243) AND ri<>3 THEN ri=4       [1189]

690 IF ta$=CHR$(240) AND ri<>2 THEN ri=1       [1829]
700 IF ta$=CHR$(241) AND ri<>1 THEN ri=2       [2020]

710 ' Position bestimmen                       [660]
720 xx=x:yy=y ' alte Position festhalten       [1834]
730 IF ri=1 THEN y=y-1:mann$=CHR$(248) ELS     [3387]
E IF ri=2 THEN y=y+1:mann$=CHR$(248)
740 IF ri=3 THEN x=x-1:mann$=CHR$(251) ELS     [4987]
E IF ri=4 THEN x=x+1:mann$=CHR$(250)
750 RETURN '    Neue Richtung bestimmt         [3364]
760 REM ************ Loeschen und Drucken       [4003]
**********************************
770 IF ri>0 THEN 800 ' Eingabe eines Befeh     [2158]
ls erhalten
780 LOCATE x,y:PEN 1:PRINT CHR$(249);          [930]
790 RETURN ' Mann bewegt sich nicht            [3552]
800 GOSUB 980                                  [1008]
810 bi(yy,xx)=5                                [585]
820 LOCATE xx,yy:PEN 3:PRINT strich$;          [2238]
830 LOCATE x,y:PEN 1:PRINT Mann$;              [1032]
840 RETURN ' Drucken beendet                   [2084]
850 REM ********** Wiederholen der Spielst     [3371]
aerke *****************************
860 ENV 1,100,1,3:ENT 1,100,5,3:SOUND 1,28     [2262]
4,300,1,1,1
870 IF x>40 THEN x=40 ELSE IF x<1 THEN x=1     [3881]
 ELSE IF y>23 THEN y=23
880 IF y<1 THEN y=1                            [770]
890 GOSUB 980                                  [1008]
900 LOCATE xx,yy:PEN 3:PRINT STRICH$;          [2238]
910 LOCATE x,y:PEN 3:PRINT CHR$(238);          [1665]
920 leben=leben-1:IF leben<0 THEN 1060         [1527]
930 x=20:y=11:pz=0:ri=0:ra=0:strich$=CHR$(     [2545]
249)
940 GOTO 190                                   [407]
950 REM ************* Fehlerroutine ******     [3404]
**********************************
960 IF ERR=4 THEN RESTORE:RESUME               [3381]
970 PRINT ERR:END                              [346]
980 REM *********** Bestimmt die Art der L     [3060]
inie ******************************
990 IF ra=ri AND ra<3 THEN strich$=CHR$(14     [2683]
9):RETURN
1000 IF ra=ri AND ra>2 THEN strich$=CHR$(1     [1231]
54):RETURN
1010 IF (ra=1 AND ri=4) OR (ra=3 AND ri=2)     [3481]
 THEN strich$=CHR$(150):RETURN
1020 IF (ra=2 AND ri=4) OR (ra=3 AND ri=1)     [3507]
 THEN strich$=CHR$(147):RETURN
1030 IF (ra=4 AND ri=1) OR (ra=2 AND ri=3)     [4853]
 THEN strich$=CHR$(153):RETURN
1040 IF (ra=4 AND ri=2) OR (ra=1 AND ri=3)     [3422]
 THEN strich$=CHR$(156)
1050 RETURN ' Linie Bestimmt                   [2029]
1060 REM *********** Programm beenden ? *      [4716]
**********************************
1070 FOR Q=1 TO 3000:NEXT                      [1230]
1080 CLS:PEN 2:LOCATE 1,2                      [804]
1090 PRINT "PUNKTE: ",PUNKTE                   [2017]
1100 PRINT "STERNE: ",ST                       [1600]
1110 PRINT "LEBEN: ",0                         [1101]
1120 PRINT "SPIELSTAERKE:";SP                  [2025]
1130 LOCATE 5,12:PEN 1                         [580]
1140 PRINT "Neues Spiel ?  (Ja oder Nein)"     [3687]
1150 qq$=INKEY$:IF LEN(qq$)=0 THEN 1150        [1668]
1160 IF qq$="J" OR qq$="j" THEN RUN ' neu      [2672]
starten
1170 IF qq$="N" OR qq$="n" THEN CLS:END EL     [942]
SE 1150
1180 DATA 226,50,1,227,25,2,229,10,3,231,5     [1356]
,4
1190 DATA 229,10,3,231,5,4,227,25,2,231,5,     [1553]
4
```

Kommen - Sehen - Kaufen

Unter diesem Motto stand die Hobbytronic und Computerschau '86 in Dortmund. Diese Computer-Ausstellung ist zwar eine der kleinsten überhaupt, läßt aber am recht hohen Stellenwert keinen Zweifel.

Einmal im Jahr freuen sich Hersteller und Anwender auf die fünf Tage dauernde Computerschau in der Dortmunder Westfalenhalle. Die Anwender deshalb, weil sich hier die Gelegenheit bietet, einen direkten und persönlichen Kontakt zu Herstellern und Vertreibern von Computerprodukten aufzubauen und Informationen aus erster Hand zu erhalten. Für den privaten Anwender präsentiert sich die Dortmunder Computerschau quasi als Messe zum Anfassen – ein sehr wichtiges Kriterium, betrachtet man einmal die sich oft steril und übermächtig darstellenden Messen in Hannover, München oder Frankfurt, wo der private Anwender auch dementsprechend angesehen wird.

Diese Feststellung läßt sich zum einen aus den zahlreichen Gesprächen ableiten, die wir beim Besuch in Dortmund mit vielen unserer Leser geführt haben. Zum anderen präsentiert sich im Gegensatz zu übrigen Ausstellungen ein völlig anderer Messecharakter, der unserer Überschrift "Kommen, Sehen, Kaufen" durchaus gerecht wird.

Dementsprechend setzen sich auch die Aussteller zusammen, deren oberstes Gebot hier einmal nicht die reine Präsentation neuer Produkte ist, sondern möglichst viel zu verkaufen, (oftmals zu Schleuderpreisen) um damit u.a. auch die Lager zu entleeren.

Der regionale Messecharakter vermittelt hier das Gefühl einer gewissen Zugehörigkeit; die Aussteller treffen auf die Basis, die sie über ihre Produktwerbung oft nur unpersönlich erreichen.

Daß dabei selbstverständlich auch das eine oder andere ungemütliche Gespräch zustande kommt, scheint allerdings die wenigsten Aussteller zu kümmern. Information und Kundendienst werden groß geschrieben und direkt vermittelt – eigentlich der Idealzustand – oder?

Die Besucherzahlen geben für die Zukunft keinerlei Bedenken; vor allem an den Wochenendtagen herrschte riesiger Andrang auf der Computerschau und der parallel stattfindenden Hobbytronik (trennen lassen sich diese beiden Ausstellungen nur schwerlich, da die angebotenen Produkte oft zu beiden Bereichen gehören).

Aussteller

Was sich auf den vergangenen Computermessen bereits abzeichnete, wurde in Dortmund zur Gewißheit. Fast alle Firmen haben ihre Produktpalette zum großen Teil auf die Schneider-Computer abgestellt, insbesondere profitiert der Bereich Software davon. Wie in Dortmund eindrucksvoll demonstriert, ist die Software-Palette für CPC bzw. Joyce inzwischen auf ein riesiges Potential angewachsen.

Den mit Abstand größten Messestand hatte die Fa. Data Media aufzuweisen, hier wurden die Computerspiele zu echten Schleuderpreisen unter die Leute gebracht.

Das Softwarehaus GFA (früher Integral Hydraulik) zeigte die bewährten Anwenderprogramme RH-Dat, Mica und RH-Büro, die auch im kleinen Ausstellungsrahmen ihre Wirkung nicht verfehlten.

Ein wahres Sammelsurium von Ausstellern tummelte sich an den beiden Ständen von Horten und Karstadt. Dort konnten u.a. Vortex und Star Division ihre Produkte vorstellen und den interessierten Anwendern Rede und Antwort stehen.

Übersichtlicher war dagegen der Literaturbereich, wo die Verlage Hüthig, Vieweg und Hofacker die Gelegenheit zur Präsentation nutzten. Der Bereich Peripherie stand eindeutig im Hintergrund – lediglich die fischertechnik computing-Serie wäre hier erwähnenswert.

Perspektiven

Der Rundgang durch die beiden Messehallen war schnell vorbei. Das gezeigte Angebot konnte in Bezug auf Exclusivität und Neuentwicklung niemanden vom "Hocker" reißen.

Die Firmen, die Produkte für den CPC bzw. Joyce anbieten, waren durchweg mit den Verkäufen zufrieden; Händlergeschäfte blieben die Ausnahme.

Mit Blick in die Zukunft läßt sich die Marktentwicklung im Bezug auf Schneider als sehr positiv bezeichnen, denn ernsthafte Konkurrenz ist nicht in Sicht. Zudem bestreitet bei Schneider niemand mehr, daß an einem neuen Computer mit exzellenten Leistungsdaten gearbeitet wird, und das läßt für die Zukunft alle Beteiligten guter Dinge sein. Vor Beginn des Weihnachtsgeschäftes dürfte allerdings nicht mit einem neuen Schneider Computer zu rechnen sein.

Neben dem CPC 464 und CPC 6128, die nach wie vor in der Käufergunst ganz oben angesiedelt sind (kein Wunder, bei diesen, auch im härtesten Einsatz, hervorragenden Computern), entpuppt sich der Joyce als Publikumsliebling, was auch die bisher erreichten Verkaufszahlen von mehr als 25.000 Stück belegen. Schneider erwartet noch eine deutliche Steigerung bis Ende des nächsten (!) Jahres.

Die starke Nachfrage macht es möglich, daß der Joyce auch im nächsten Jahr deutliche Erfolge erzielen wird. Dieser Trend läßt sich u.a. auch in der Verkaufshitparade laut Chip ablesen; innerhalb kürzester Zeit schoß der Joyce von 0 auf Platz 2 – der Aufsteiger des Monats!

Fazit

Durch den bereits beschriebenen Messecharakter stellt die Dortmunder Computerschau ein ganz besonderes Ereignis in der sonst sehr pompösen Messelandschaft dar. Die Anwender fühlen sich hier ganz besonders gut aufgehoben, weil auch das Umfeld gezielt einen persönlichen Bezug aufweist. Sensationen können auch in Zukunft nicht auf der Computerschau erwartet werden, dafür konzentrieren sich die großen Hersteller zu sehr auf andere Ausstellungen. Doch vielleicht kommen auch die Branchenriesen einmal zu der Überzeugung, daß der Dienst am Kunden auch in Form einer Beratungsstelle durchaus auf kleinen Messen seinen Zweck erfüllen könnte... (SR)

Bestellkarte »LESERSERVICE«

Abgesandt am:

______________________ 198__

Bemerkungen:

____

____

____

____



Bestellkarte »LESERSERVICE«

Absender: *(Bitte genaue Anschrift angeben!)*

____
Name

____
Vorname

____
Firma

____
Straße/Nr./Postfach

____
PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

Schneider CPC International
Postfach 250

3440 Eschwege

Kleinanzeigen

– Die preiswerte Möglichkeit Ihre Angebote an den/die richtige(n) Mann/Frau zu bringen!

Private Kleinanzeigen:
5,– DM/Zeile (inkl. MwSt.)

Geschäftliche Kleinanzeigen:
7,– DM/Zeile (zuzgl. MwSt.)



»Klein-Anzeigen-Markt«

Absender: *(Bitte genaue Anschrift angeben!)*

____
Name

____
Vorname

____
Firma

____
Straße/Nr./Postfach

____
PLZ/Ort

Bitte ausreichend frankieren

Antwortkarte

Schneider CPC International
Postfach 250

3440 Eschwege

Data Media »COMPUTERCLUB«

Abgesandt am:

______________________ 198__

Bemerkungen:

____

____

____

____

Bitte senden Sie mir folgende Artikel aus dem ☐ Mailorder Angebot ☐ Clubanga

☐ Ich möchte weitere Informationen über den Data Media Computerclub, ehe ich mich einer Mitgliedschaft entschließe (2,– DM in Briefmarken habe ich beigelegt).
Mein Name und meine Anschrift lautet:

Anzahl	Best.-Nr.:	Titel	Preis
		Club-Beitrag	60,– DM
		Porto + Verpackung	3,– DM
		Gesamtbetrag	

*falls gewünscht, bitte Anzahl angeben

☐ Den vorstehenden Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.
☐ Ich bitte um Lieferung per NN, zuzügl. der Nachnahmegebühr (nur innerhalb der BRD).

Datum ______ Unterschrift (bei Minderjährigen: Unterschrift des gesetzl. Vertreters)

data media gmbh

Bitte ausreichend frankieren

Data Media GmbH
– Mailorder/Computerclub -
Ruhralle 55

4600 Dortmund

»Leserservice«

Ich mache von Ihrem Angebot Gebrauch und bestelle hiermit:

☐ **GAC** – das Traumtool für den Abenteuerfreund.
Mit komplettem deutschen Handbuch.
für Schneider CPC 464/664/6128 — Stück .... ☐ Kass. à 85,– DM ______

Stück .... ☐ Disk. à 95,– DM ______

☐ **"Goldene 7"** -1- – die Qualitätssoftware von Lesern für Leser.
Mit sieben Aktions- und Abenteuer-Spielen.
für Schneider CPC 464/664/6128 — Stück .... ☐ Kass. à 20,– DM ______

Stück .... ☐ Disk. à 30,– DM ______

Porto/Verpackung 3,– DM

Gesamtbetrag ______

☐ Den vorstehenden Betrag bezahle ich mittels des beigefügten Verrechnungsschecks.
☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum — Unterschrift (bei Minderjährigen Unterschrift des gesetzl. Vertreters)

»Kleinanzeigen-Markt«

Private Anzeigen: Nur DM 5,- je angefangene Zeile, inkl. gesetzlicher Mehrwertsteuer.
Geschäftliche Empfehlungen: DM 7,- je angefangene Zeile, zzgl. gesetzlicher Mehrwertsteuer.
Kreuzen Sie bitte an, in welche Rubrik (s. Karte) Ihre Anzeige gehört, schreiben Sie Ihren Text in die Karte (jedes Kästchen = ein Zeichen; Buchstabe, Satzzeichen oder Wortzwischenraum).
Absender und Unterschrift auf die Karte und heraustrennen.
Achtung! Der Abdruck erfolgt **nur** gegen Vorkasse (Verrechnungsscheck).
Bitte veröffentlichen Sie meine Anzeige in der nächsterreichbaren **„Schneider CPC International"** für
☐ private Zwecke ☐ gewerbliche Zwecke.
Das ist der Text:

☐ Die Anzeige soll als Chiffre-Anzeige erscheinen (nur möglich bei Privat-Anzeige).
Chiffre-Gebühr: DM 10,- inkl. MwSt. zzgl. zum Anzeigenpreis.

In dieser Rubrik:
☐ **Biete an** — ☐ Hardware — ☐ Software
☐ **Suche** — ☐ Hardware — ☐ Software
☐ **Tausch**
☐ **Stellenmarkt/freie Mitarbeit**
☐ **Geschäftsverbindungen**
☐ **Verschiedenes**

»Data Media Computerclub«
Beitrittserklärung

Ich möchte von den günstigen Einkaufsangeboten des Data Media Com[-]puterclubs Gebrauch machen und beantrage hiermit die Mitgliedschaft [ab] 1.5.1986 zunächst für ein Jahr. Die Mitgliedschaft soll sich um jeweils e[in] weiteres Jahr verlängern, wenn ich nicht 3 Monate vor Ablauf des Zeitra[u]mes kündige.
Ich kann ab sofort aus dem Club-Angebot einkaufen und alle weiter[en] Vorteile des Clubs nutzen. Der Clubausweis wird mir nach Zahlung d[es] Jahresbeitrages von 60,– DM zugesandt.

Vor- und Zuname: ______
Straße/Haus-Nr.: ______
Postleitzahl — Wohnort ______
Beruf: ______
Geburtsdatum: ______ Telefon ______

Datum — 1. Unterschrift (bei Minderjährigen Vor- u. Zuname d. ges. Vertr.)

Ich kann eine Woche nach Erhalt der Mitgliedsbestätigung diesen Beitritt schriftlich widerrufen. Der Widerruf ist zu richten an: Data Media GmbH, Computerclub, Ruhrallee 55, 4600 Dortmund. Ich bestätige durch meine 2. Unterschrift, daß ich auf dieses Widerrufsrecht aufmerksam gemacht worden bin.

Datum — 2. Unterschrift (bei Minderjährigen Vor- u. Zuname d. ges. Vertr.)

»Leserservice«

Ausgewählte Software für unsere Leser!
Heute bestellt – morgen geliefert!

»Kleinanzeigen-Markt«

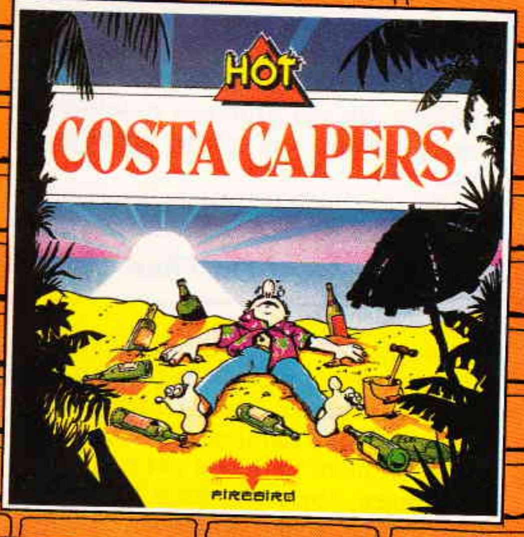
Abgesandt am:

______ 198___

Bemerkungen:

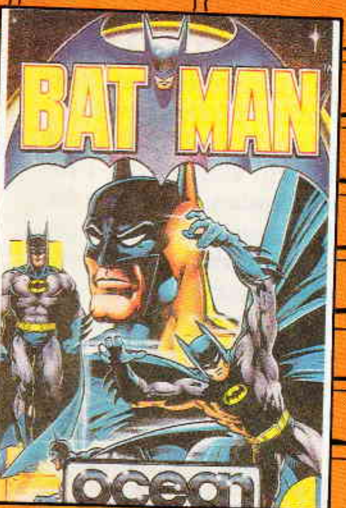
______

Data Media Computerclub









Zu den Abbildungen:

Art.-Nr.		Titel	Verk. Preis (Club)	Verk. Preis
188	C	Costa Capers	29,–	39,–
189	C	Runestone	29,–	39,–
191	C	The Way of the Tiger	29,–	39,–
331	D	The Way of the Tiger	45,–	55,–
193	C	Ping Pong	29,–	39,–
192	C	Movie	29,–	39,–
195	C	Batman	29,–	39,–
199	C	They sold a Million (2)	29,–	39,–
194	C	Strike Force Harrier	29,–	39,–
333	D	Strike Force Harrier	49,–	59,–
187	C	Spindizzy	29,–	39,–
355	D	Spindizzy	49,–	59,–

C = Cassette
D = 3"-Diskette

und hier weitere Spitzen-Spiele:

Art.-Nr.		Titel	Verk. Preis (Club)	Verk. Preis
166	C	Bombjack	25,–	39,–
327	D	Bombjack	39,–	49,–
167	C	Commando	29,–	39,–
328	D	Commando	45,–	55,–
184	C	Saboteur	29,–	39,–
185	C	Turbo Esprit	29,–	39,–
186	C	Fairlight	29,–	39,–
190	C	Thing on a Spring	29,–	39,–
196	C	Doomsday Blues	29,–	39,–
197	C	Monopoly	39,–	49,–
332	D	Monopoly	49,–	59,–
222	C	Bounder	29,–	39,–
357	D	Bounder	39,–	49,–
198	C	Get Dexter	29,–	39,–
200	C	Alien Highway	29,–	39,–
201	C	Moon Cresta	29,–	39,–
356	D	Cluedo	49,–	59,–

Schwarze Preise: Verkaufspreise für Nicht-Clubmitglieder
Rote Preise: nur für Clubmitglieder!









Data Media Mailorder und Computerclub

Bestellen Sie noch heute, Lieferung erfolgt umgehend! Zahlung per Vorkasse oder Nachnahme zuzügl. Porto- bzw. Nachnahmegebühr (Nachnahme ins Ausland ist nicht möglich).

Weitere Artikel in unserem Club- oder Mailorderkatalog. Bitte anfordern (2,– DM für Rückporto beilegen).
Data Media GmbH, Ruhralle 55, 4600 Dortmund, Tel.: (02 31) 12 50 71-3

MAIL

Bitte Clubhinweis auf Seite 117 beachten!

Bitte Bestellkarte benutzen!

Ton und Bild zum TV über (Scart)-Buchse



"Ich hab's!"- das war der ganze Inhalt des Telegramms, welches uns erreichte. Die Unterschrift stammte eindeutig von Prof.-Dr. Schnösel Sägezahn. Nach der Implantation des achten Bit in den CPC (Heft 5) war der Chef nicht davon abzubringen, daß Großartiges unserer harren mußte. Also: nichts wie hin. In der Computerklinik angekommen, läßt man uns in die heiligen Hallen des Professors vor; dieser heißt uns aufs Herzlichste willkommen. Dann kommt er zur Sache - und das nicht zu knapp.

Die Entdeckung

Schnösel-Sägezahn hatte es in der Tat geschafft, neben dem Bild auch den Stereo-Ton über den SCART-Eingang vom CPC zum TV zu übertragen. Da wir unsere Ungläubigkeit nicht unterdrücken konnten, trat der Professor alsbald den praktischen Beweis an. Der Testaufbau war nach Art des Professors völlig einfach. Ein CPC war über ein einziges Kabel mit einem stereotüchtigen Fernsehgerät über dessen SCART-Buchse verbunden. Erst kurz vor dem Computer wurde dieses Kabel durch eine Kupplung geteilt und die geteilten Anschlüsse der Monitor- bzw. der E/A-Buchse zugeführt. Sägezahn trat zu seiner Apparatur und begann die Vorführung, wobei uns einige Einzelheiten auffielen, die der Professor in seiner Vergeßlichkeit gar nicht erwähnt hatte.

Da kriegst Du Töne

„Obwohl der im CPC eingebaute Lautsprecher nicht abgeschaltet wird, werden die beim Laden von Kassette auftretenden Geräusche dennoch nicht in Stereo auf das TV übertragen, ebenso können Sie den CPC-Lautsprecher, wie bisher, lauter und leiser stellen", erklärt Sägezahn, „außerdem hat sich im Lauf der langen Tests ein Arbeitsabstand zum Bildschirm von 2 - 3 Metern bewährt." Nachdem wir nach Vorschrift Platz genommen hatten, startete der Professor seine Aufbauten, und siehe da: wir wurden nicht nur von einem gestochen scharfen, farbigen Bild angestrahlt, auch umschmeichelte ein glasklarer, voller Stereo-Ton unsere Ohren. Es war phantastisch und - es sollte noch besser kommen.

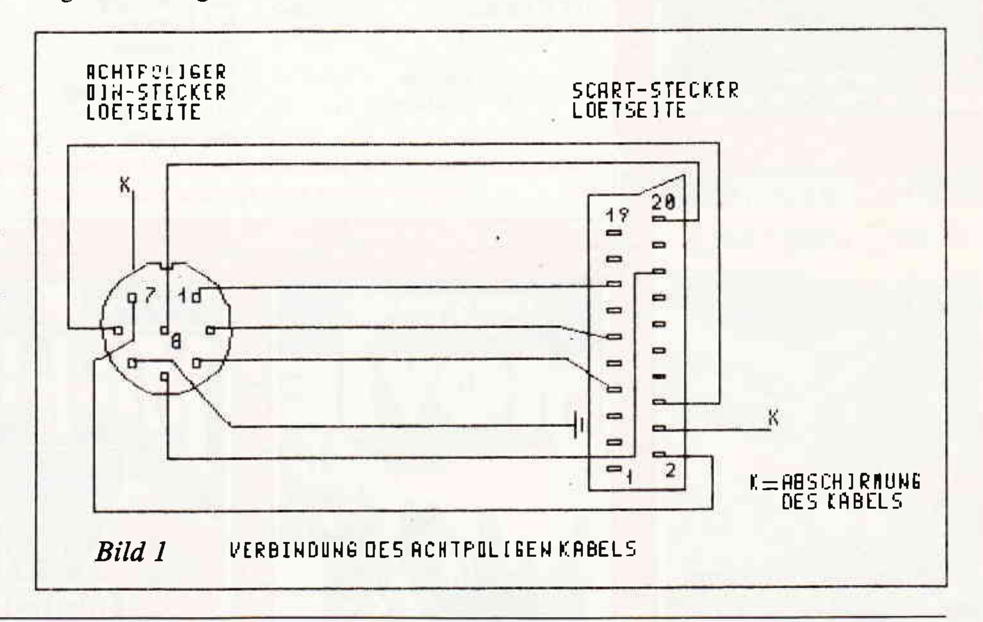


Bild 1 VERBINDUNG DES ACHTPOLIGEN KABELS

Noch mehr Töne

Durch genaues Einpegeln des CPC-Lautsprechers auf die Lautstärke des TV-Gerätes ließ sich der Sound nochmals verbessern. Als der Professor über den fünfpoligen DIN-Stecker eine bereitstehende Stereoanlage an den TV-Apparat anschloß, umfing uns ein wahrhaft raumfüllender Sound. Sofort bestürmten wir Sägezahn, uns die Bauanleitung dieser Erweiterung zu überlassen, was er, allerdings unter der Ermahnung, die Erfindung nur zu friedlichen Zwecken zu nutzen, auch tat. So können wir Ihnen völlig exklusiv den Bericht "Bild und Ton zum TV" anbieten. Hier nun die Bauanleitung.

So wird es gemacht

Benötigtes Material:
ein Euro AV (SCART)- Stecker
ein sechspoliger DIN- Stecker
ein Klinkenstecker 3,5 mm Stereo
ein achtpoliger DIN- Stecker
eine achtpolige DIN- Buchse
30 cm abgesch. Kabel, 6-adrig
30 cm abgesch. Kabel, 2-adrig
beliebig abgesch. Kabel, 8-adrig

Benötigtes Werkzeug:
Lötkolben max. 30 W
scharfes Messer
Seitenschneider
Elektroniklötzinn

Mit dem Messer wird die Ummantelung des achtadrigen Kabels so weit entfernt, wie es der Stecker erfordert, anschließend werden die einzelnen Adern abisoliert. Nun werden diese Adern gemäß Bild 1 an die Lötpunkte des DIN-Steckers einerseits und an die des SCART-Steckers andererseits angelötet. Die Kabelabschirmung wird

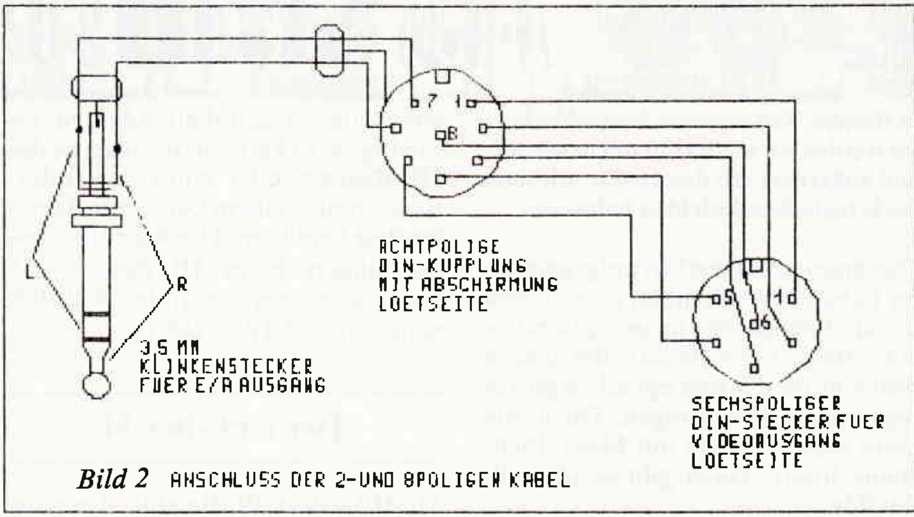


Bild 2 ANSCHLUSS DER 2-UND 8POLIGEN KABEL

auf Pin 4 des SCART-Steckers und auf das Gehäuse des DIN-Steckers gelötet. Diese Verbindung darf nicht vergessen werden, weil ohne sie eine saubere Tonübertragung nicht möglich ist. Pin 5 des DIN-Steckers verbindet man mit der Masse des SCART-Steckers. Diese besteht aus einem separaten Anschluß, der mit den Klammern an der Außenseite des Steckers verbunden ist. Vor dem Schließen der Stecker ist darauf zu achten, daß alle Lötstellen in Ordnung sind und diese nicht gequetscht werden oder die Abschirmung berühren.
Nun wird die Ummantelung des zweiadrigen Kabels an den Enden entfernt, die Adern werden abisoliert und gemäß Bild 2 verlötet.
Auch hier muß die Abschirmung des Kabels mit den Gehäusen von Klinkenstecker und DIN-Kupplung verbunden werden.
Auf gleiche Weise wird mit dem sechsadrigen Kabel nach Bild 2 verfahren. Sollten Sie ein TV-Gerät der Firma Philips besitzen, können Sie auf die Verbindung von Pin 4 des sechspoligen DIN-Steckers zu Pin 4 der achtpoligen Kupplung verzichten, da diese Geräte keine Synchronisation benötigen. Nun werden die Stecker unter Berücksichtigung der oben erwähnten Vorsichtsmaßnahmen geschlossen – fertig.
Ist eine Gesamtkabellänge bis max. ca. 2 m geplant, so ist es vollkommen ausreichend, das achtadrige Kabel direkt zum Klinken- und sechspoligem DIN-Stecker zu führen; die achtpolige Steckverbindung soll bei größeren Kabellängen lediglich ein Kabelwirrwarr vermeiden.

Das erwartet uns demnächst

Nachdem Prof.-Dr. Schnösel-Sägezahn den Aufbau seiner Bild- und Tonübertragung detailliert erläutert hatte, wies er auf einige noch zu erwartende Entwicklungen hin.
So wird zur Zeit an einer Bild- und Tonübertragung zu TV-Geräten ohne SCART-Buchse gearbeitet, so daß auch die Besitzer dieser Geräte nicht verzweifeln müssen. Weiterhin ist ein Terminal mit Bild- und Tonübertragung, eingebautem Netzteil, Taktgebern, Videoeinspeisung und anderem mehr in Vorbereitung. Auch auf diese Hardwareerweiterungen dürfen wir also gespannt sein und wünschen zunächst allzeit sauberes Bild und guten Ton. (R. Nitsche/ME)

Z-80 Assemblerkurs

In diesem Teil unseres Assemblerkurses werden wir uns weiter mit den 8-Bit- und außerdem mit den 16-Bit-arithmetisch/logischen Befehlen befassen.

Die meisten Assemblerprogrammierer haben vor Assembler eine andere Computersprache kennengelernt – etwa Basic oder Pascal. Bei diesen problemorientierten Sprachen gibt es sogenannte IF-Abfragen. Doch wie kann man so etwas auf Maschinenebene lösen? Dazu gibt es die CP-Befehle.
Allerdings kann hierdurch nur verglichen werden. Die Reaktion auf den Vergleich muß mit einem zweiten Befehl erfolgen – also ganz im Gegensatz zu Basic oder Pascal.

Beispiel:

Basic	Assembler
10 A = 1	LD A,1
20 A = A + 1	LOOP ADD 1
30 IF A<>10 THEN 20	CP 10
	JR NZ,LOOP

In diesem Beispiel stehen sich jeweils die Zeilen gegenüber, die für dieselbe Anweisung stehen.
Die CP-Anweisung funktioniert zum einen mit einem angegebenen Register, wobei die Ausführungszeit vier Taktzyklen dauert. Hierbei ist es möglich, das A-, B-, C-, D-, E-, H- und L-Register zu verwenden.
Zum anderen ist die Angabe einer 8-Bit-Konstanten möglich. Die Ausführungszeit verlängert sich wegen der einzulesenden Konstanten auf sieben Taktzyklen.
Außerdem können natürlich auch die Indexregister plus einem 8-Bit-Offset oder dem Inhalt des HL-Registers als Adressangabe benutzt werden. Die Ausführung dauert bei den Index-Registern IX- bzw. IY 19 Taktzyklen, bei Benutzung des HL-Registers nur 7 Taktzyklen.

Die INC- und die DEC-Befehle

Die INC- bzw. DEC-Befehle sind bei allen mir bekannten Mikroprozessoren vorhanden. Sie bewirken folgendes:
Der INC-Befehl erhöht das angegebene Register oder die Speicherstelle um eins, der DEC-Befehl subtrahiert eins. Dabei werden natürlich alle Flags entsprechend gesetzt. Will man ein Register inkrementieren oder dekrementieren, so dauert dies vier Taktzyklen. Ist dagegen eine Speicherstelle zu erhöhen oder zu erniedrigen, so kann man entweder das HL-Register oder eines der Index-Register plus einem Offset zur Adressierung benutzen. Die Ausführungszeit dauert beim HL-Register 11 Taktzyklen, bei der Index-Registerbenutzung 23 Taktzyklen.

Der CPL-Befehl

Mit Hilfe des CPL-Befehls ist es möglich, den Akku zu invertieren, das heißt, sein Einerkomplement wird gebildet und wieder im Akku abgelegt. Der ursprüngliche Wert wird dabei überschrieben.

Beispiel:
Der Akku enthält den Wert binär 1001 1100.
Nach Ausführung des CPL-Befehls 0110 0011.
Die Ausführungszeit beträgt vier Taktzyklen.

Der NEG-Befehl

Der NEG-Befehl ist dem CPL-Befehl sehr ähnlich. Er bewirkt eine Konvertierung zum Zweierkomplement im Akku. Dies kann schriftlich auf zwei Arten erfolgen:

1. Es wird das Einerkomplement gebildet und anschließend wird der Wert um eins erhöht.
2. Der Wert wird von Null abgezogen.

In unserem Beispiel würde das dann so aussehen:
Der Akku enthält den Wert binär 1001 1100
Nach Ausführung des NEG-Befehls 0110 0100 (= 01100011 + 1)
Die Ausführungszeit beträgt wegen der Mehrarbeit acht Taktzyklen.
Nun gibt es noch einen ganz besonderen Befehl. Er heißt DAA. Er ist insoweit etwas Besonderes, als daß er nur dann Verwendung findet, wenn in den Registern mit Dezimalzahldarstellung (gepackte BCD-Darstellung) gearbeitet wird. In diesem Falle wird ein Nibble (= ein halbes Byte bzw. vier Bits) benutzt, um eine Dezimalziffer (0 – 9) zu speichern. Deshalb kann jedes (8-Bit-)Byte zwei BCD-Ziffern speichern. Beispielsweise wird die Zahl 13 folgendermaßen dargestellt.

```
   1    3
0001 0011
```

Mit dieser Zahl kann nun natürlich nicht binär weitergerechnet werden, denn binär würde die Zahl nicht 13, sondern 19 bedeuten. Sie merken schon, bei dieser Art Darstellung gibt es einige Probleme. Wird jetzt zu dieser Zahl eine 29 (BCD dargestellt) addiert, so ergibt sich folgendes:

```
  0001 0011  13
+ 0010 1001  29
---------------
  0011 1100  3C
```

Dies ist jedoch dezimal gesehen unmöglich, denn bei der BCD-Darstellung gibt es keine Zahl C. Jetzt hilft uns jedoch der DAA-Befehl weiter. Er korrigiert das Ergebnis so, daß wieder der richtige Wert entsteht. Und dies macht er auf folgende Weise.
Zuerst wird geprüft, ob das niederwertigere Nibble größer ist als neun. Sollte dies der Fall sein, so entstand eine ungültige Ziffer. In diesem Falle muß dann eine Sechs addiert werden. Die Prüfung erfolgt mit Hilfe des Statusregisters (H-Flag).
In unserem Beispiel ist das H-Flag gesetzt. Es muß also eine Sechs addiert werden. Wir wollen es probieren:

```
  0011 1100  3C
+ 0000 0110  06
---------------
  0100 0010  42
```

Wie Sie selbst sehen können, ist das Ergebnis nach Ausführung des DAA-Befehls wieder richtig. Nach Beendigung wird das obere Nibble mit Hilfe des P/V-Flags getestet und evtl. eine 60Hex addiert. In unserem Beispiel ist dies nicht der Fall, da das obere Nibble nicht größer ist als 6.

16-Bit-arithmetisch/logische Befehle

Bei den nun folgenden Befehlen ist zu beachten, daß die Flags nicht entsprechend dem Ergebnis gesetzt sind. Denn, wie Sie ja sicher wissen, ist der gute Z-80 ein 8-Bit-Prozessor. Sollen nun zwei mal 16 Bit miteinander verknüpft werden, so muß dies in zwei Schritten passieren. Deshalb sind die Flags immer entsprechend der zweiten Addition bzw. Subtraktion gesetzt. Zweite Addition bzw. Subtraktion – damit ist die Addition bzw. die

Subtraktion des höherwertigen Bytes gemeint. Allerdings ist dies meistens nicht, oder nur ganz selten, verwendbar.

Der 16-Bit ADD-Befehl

Bei dem 16-Bit ADD-Befehl wird das Ergebnis im HL-Register oder in einem der beiden Index-Register abgelegt. Bei dem 8-Bit ADD war das Ergebnis nach der Ausführung immer im Akku. Da der Akku aber nur 8 Bit groß ist, kann ein 16-Bit-Ergebnis darin nicht abgelegt werden. Deshalb wird hier das HL-Register oder eines der beiden Index-Register benutzt.

ADD HL,rr

Dieser Befehl wird dazu verwendet, eines der Registerpaare BC, DE, HL oder SP zu HL zu addieren. Die Ausführungszeit dauert 11 Taktzyklen.

Mit Hilfe dieses Befehls ist es beispielsweise auch möglich, das HL-Register mit z.B. 8 zu multiplizieren, ohne ein anderes Doppelregister zu verwenden.

Und dies funktioniert so:

Eine Multiplikation mit 8 kann auch verwirklicht werden, wenn man den Wert dreimal mit sich selbst addiert. In Assembler könnte dies auf folgende Weise erreicht werden.

```
ADD HL,HL   ;2-facher Wert in HL
ADD HL,HL   ;4-facher Wert in HL
ADD HL,HL   ;8-facher Wert in HL
```

Eine so programmierte Multiplikation mit 8 ist natürlich nicht so flexibel, da immer mit 8 multipliziert wird. Es ergibt sich jedoch eine bessere Laufzeit und weniger Speicherbedarf, als es bei der Benutzung eines externen Unterprogramms, das 16-Bit-Multiplikationen vornimmt, der Fall wäre. Wägen Sie deshalb immer genau ab, welche Verfahrensweise günstiger ist.

Ein ähnlicher Befehl ist:

ADC HL,rr

Hierbei passiert das gleiche, mit dem Unterschied, daß das Carry-Flag noch hinzuaddiert wird. Die Ausführungszeit verlängert sich dadurch auf 15 Taktzyklen.

Ebenfalls sehr ähnlich arbeitet:

SBC HL,rr

Dieser Befehl subtrahiert das angegebene Registerpaar plus dem Carry-Flag vom HL-Register. Die Ausführungszeit beträgt, genau wie beim ADC-Befehl, 15 Taktzyklen.

Beachten Sie hierbei, daß es nicht möglich ist, ohne Carry zu subtrahieren.

Es gibt noch weitere 16-Bit-Additionsbefehle – die Additionsbefehle mit dem Index-Register, allerdings ohne Offset.

ADD IX,rr bzw. **ADD IY,rr**

Zu bachten ist dabei nur, daß das HL-Register nicht benutzt werden kann. Stattdessen ist entweder das IX- oder das IY-Register im Spiel. Die Ausführungszeit dauert ebenfalls 15 Taktzyklen.

Bei den 16-Bit-arithmetisch/logischen Befehlen gibt es noch zwei weitere Gruppen. Zum einen ein INC rr und zum anderen ein DEC rr.

Bei beiden Befehlsgruppen können folgende Doppelregister verwendet werden:

BC, DE, HL, SP, IX und **IY**

Die Ausführungszeit dauert normalerweise sechs Taktzyklen, bei Benutzung der Index-Register zehn Taktzyklen.

In den nächsten Teilen unseres Kurses gehen wir noch auf die Rotations- und Schiebebefehle, die Befehle zur Einzelbitbearbeitung, die CPU-Steuerbefehle, die Sprungbefehle und die Ein-/Ausgabebefehle ein. Danach wird unser Kurs beendet sein, und wir hoffen schon jetzt, daß Sie danach in der Lage sind, alle gewünschten Problemstellungen in Assembler zu lösen.

Bis zum nächsten Kurs.

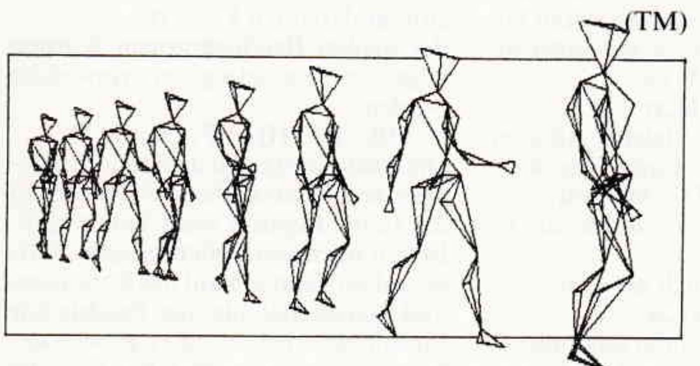
(HF)

3D-Läufer

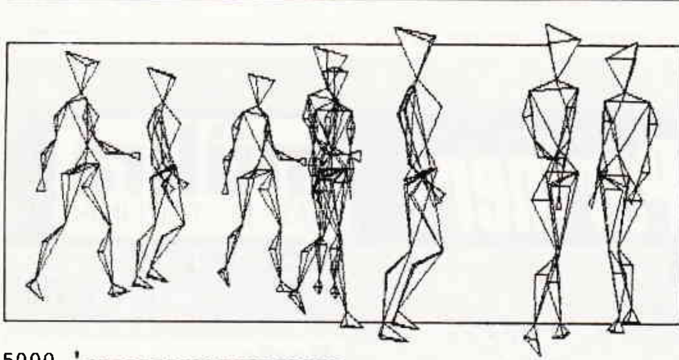
Nachtrag

Folgende kleine Routinen können Sie in Ihr Basiclisting einsetzen.
Diese Routinen stellen reine Demonstrationen dar, die komplizierte Bewegungsabläufe und Laufbahnen erzeugen.
Haben Sie selbst einen interessanten Bewegungsablauf geschrieben, so teilen Sie uns das bitte mit. An schönen Demos sind wir immer interessiert.

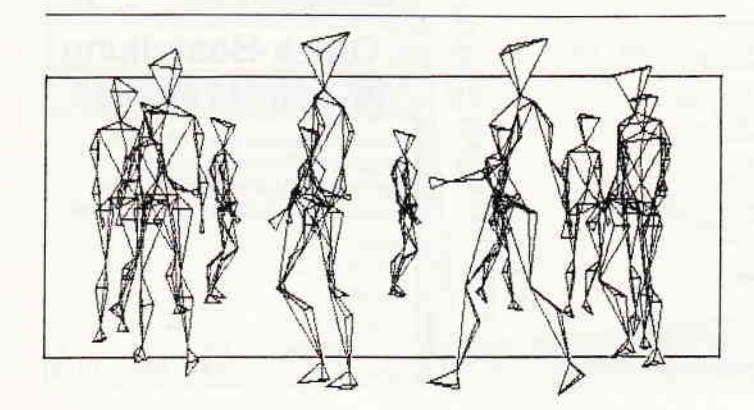
(TM)



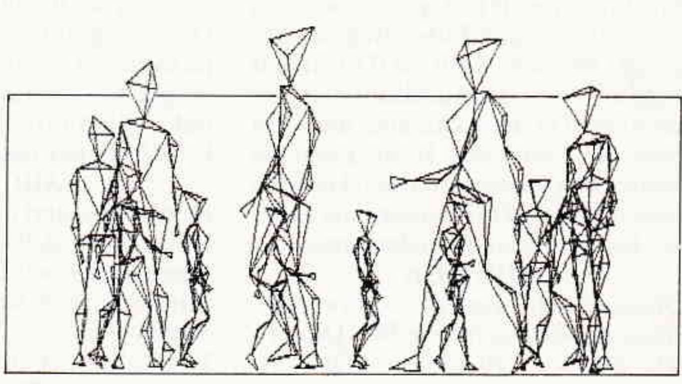
```
5000 '--------------------
5010 '          LAUFBAHN
5020 '--------------------
5030 VX=VX-7
5040 VY=VY-8
5050 GOTO 1030
```



```
5000 '--------------------
5010 '          LAUFBAHN
5020 '--------------------
5060 LSTEP=45:R=LSTEP/7
5065 IF AL>=360 THEN AL=AL-360
5066 IF AL>=180 THEN 5067 ELSE 5070
5067 VX=VX+SIN(AL)*R
5068 VY=VY+COS(AL)*R
5069 AL=AL+LSTEP:GAMMA=GAMMA-LSTEP:GOTO 1030
5070 VX=VX-SIN(AL)*R
5080 VY=VY+COS(AL)*R
5090 AL=AL+LSTEP:GAMMA=GAMMA-LSTEP:GOTO 1030
```



```
5000 '--------------------
5010 '          LAUFBAHN
5020 '--------------------
5060 LSTEP=30:R=LSTEP/3.5
5070 VX=VX-SIN(AL)*R
5080 VY=VY+COS(AL)*R
5090 AL=AL+LSTEP:GAMMA=GAMMA-LSTEP:GOTO 1030
```



```
190 '--------------------------------
200 'POSITIONIERUNG FLUCHTPUNKT/FIGUR,GROESSE,DEHN
UNG
210 '--------------------------------
220 ALPHA=90:BETA=0:GAMMA=90
230 VX=16:VY=50:VZ=10
240 AB=4:E=200
250 ORIGIN 0,-170
```

```
190 '--------------------------------
200 'POSITIONIERUNG FLUCHTPUNKT/FIGUR,GROESSE,DEHN
UNG
210 '--------------------------------
220 ALPHA=90:BETA=0:GAMMA=90
230 VX=16:VY=50:VZ=1
240 AB=4:E=200
250 ORIGIN 0,20
```

```
190 '--------------------------------
200 'POSITIONIERUNG FLUCHTPUNKT/FIGUR,GROESSE,DEHN
UNG
210 '--------------------------------
220 ALPHA=90:BETA=0:GAMMA=90
230 VX=16:VY=50:VZ=-10
240 AB=4:E=200
250 ORIGIN 0,235
```

```
7000 '------------------------
7010 'DATEN FUER PHASE  (WINKEL)
7020 '------------------------
7030 DATA 0
7040 DATA 200,200,-17,17,18,18,-20,40,-20,-16,-70,
-70,10,10,-60
7050 DATA 0,-10,-10,0,0,0,0,0,0,0,0,0,0,0,0
7060 DATA 2
7070 DATA 200,200,-17,0,-26,-26,5,5,-60,0,-26,-26,
5,5,-60
7080 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
7090 DATA 2
7100 DATA 200,200,-17,-16,-70,-70,10,10,-60,17,18,
18,-20,40,-20
7110 DATA 0,10,10,0,0,0,0,0,0,0,0,0,0,0,0
7120 DATA 2
7130 DATA 200,200,-17,0,-26,-26,5,5,-60,0,-26,-26,
5,5,-60
7140 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
7150 DATA 2
7160 DATA 200,200,-17,17,18,18,-20,40,-20,-16,-70,
-70,10,10,-60
7170 DATA 0,-10,-10,0,0,0,0,0,0,0,0,0,0,0,0
```

Basic-Kurs

Weiter geht es in unserem Basic-Kurs mit dem dritten Modul unserer Adressverwaltung.

Dieses Modul stellt den Kern der ganzen Verwaltung dar. Es ermöglicht es, nach Adressen zu suchen und diese dann gegebenenfalls zu ändern. Das Listing des ersten Teils dieses sehr komplexen Moduls finden Sie am Ende des Kurses. Welche Funktionen benutzt werden können sowie den grundsätzlichen Aufbau, soll Ihnen diese Lektion vermitteln. Mit dem ersten Teil des Moduls können Sie schon Adressen suchen. Das Ändern der Adressen erfahren Sie in der nächsten Lektion.

Zunächst einmal der Aufbau des Moduls:

Nachdem im Hauptmenue der Programmpunkt "Suchen/Ändern" angesprochen wurde, soll der Rechner die Unterroutine ab Zeile 2000 ausführen. Dieser Teil ist bereits programmiert. Angekommen in Zeile 2000, wird mit CLS zunächst der Bildschirm gelöscht. Danach wird ein kleines Menue ausgegeben und die Tastatur abgefragt. Im Menue steht die Bezeichnung der Felder, nach denen gesucht werden kann sowie eine Möglichkeit, bei irrtümlicher Falschwahl wieder ins Hauptmenue zurückzukehren. Nach Eingabe der Feldnummer wird in Zeile 2200 eine Locateposition angegeben, und je nach Höhe der Feldnummer auf eine Eingabemaske verzweigt. Hier finden wir auch einen neuen Befehl.

ON GOTO ist eine enorme Programmiererleichterung und wird mit der Syntax:

ON wert GOTO zeilennummer,zeilennummer,...

eingegeben. Hat Wert z.B. einen Inhalt von 3, so wird nach der dritten Zeilennummer hinter GOTO gesprungen. Man hätte das Ganze auch schreiben können:

2200 IF a=1 THEN GOTO 2210
2210 IF a=2 THEN GOTO 2210
2220 IF a....

Sie werden jedoch einsehen, daß der ON-GOTO-Befehl wesentlich praktischer ist. Das gleiche funktioniert übrigens auch mit ON GOSUB.

In den Zeilen 2210 - 2320 steht unsere Eingabemaske aus der vorigen Lektion. Die Subroutine INPUT, die ja noch verbessert werden soll und ab Zeile 1510 steht, wird hier noch einmal verwendet, da alle Parameter von der Maskenroutine übergeben werden können. Der praktische Nutzen von Subroutinen wird hier sehr klar dargestellt.

In den Zeilen 2330 - 2370 steht die eigentliche Suchroutine. Hier wird zunächst einmal die Länge von SUCH$, also dem String, den wir in der Eingabemaske eingetippt haben, festgestellt. Danach kommt eine Schleife von 0 bis zur höchsten Adressnummer.

In Zeile 2350 werden von der Feldvariablen, die in Feldnr bezeichnet wird, die ersten Buchstaben in SCAN$ übertragen, und zwar in der Länge von SUCH$. Danach wird verglichen, ob SUCH$ gleich SCAN$ ist. Trifft dies zu, wird die Adresse in 2400 ausgegeben. Trifft es nicht zu, wird der Zähler erhöht und weiter gesucht. Nachdem alle Felder gescannt sind, wird eine Meldung ausgegeben und zum Suchmenue zurückgesprungen. Die Zeile 2380 enthält den CALL & BB06. Dies ist eine Schneider Firmware-Routine, die auf einen Tastendruck wartet. Der Befehl hat mit Basic eigentlich nichts zu tun, sondern fällt schon in den Bereich MC-Programmierung. Wir verwenden ihn aber trotzdem, weil er so schön komfortabel ist.

Ab 2400 wird die gefundene Adresse komplett ausgegeben, und ein kleines Menue zur Verfügung gestellt, in dem abgefragt wird, ob man Ändern, Weitersuchen oder ins Menue zurückkehren möchte. Diese Routine benötigt keine weitere Erklärung.

Den Menuepunkt Ändern können Sie anwählen, erhalten jedoch nur eine Fehlermeldung, da der entsprechende Teil in 2400 noch nicht programmiert ist.

Denken Sie daran, daß das Modul ohne die vorhergehenden Teile des Kurses nicht läuft, da die Routine INPUT und die Adressen fehlen. Sie sehen, daß die eigentliche Suchroutine nur vier Zeilen lang ist. Dadurch wird das a: Suchen sehr schnell und b: wieder einmal bewiesen, daß Basic doch eine recht komfortable Sache ist.

Unsere Adreva bekommt langsam Hand und Fuß. Der "Schnickschnack" mit Menues und so weiter, ist nur deshalb so lang, damit das Ganze nachher auch ordentlich aussieht. Viel Spaß mit dem neuen Modul.

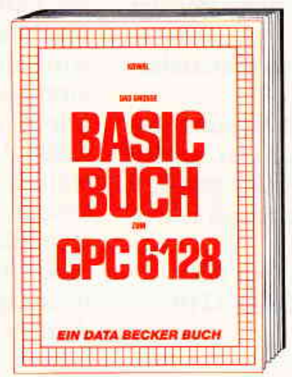
(TM)

```
2000 CLS
2010 PRINT" ADRESSEN  SU HEN / AENDERN"
2020 PRINT"----------------------------------------
"
2030 LOCATE 2,4:PRINT"[1] NAME              [2] VORNAM
E"
2040 LOCATE 2,6:PRINT"[3] STRASSE           [4] HAUSNU
MMER"
2050 LOCATE 2,8:PRINT"[5] PLZ               [6] WOHNOR
T"
2060 LOCATE 2,10:PRINT"[7] TEL.NR.           [8] BEMER
KUNG"
2070 LOCATE 2,12:PRINT"            [9] HAUPTMENUE"
2080 PRINT:PRINT"-----------------------------------
------"
2090 PRINT"BITTE SUCHFELD WAEHLEN [ 1 - 9 ]"
2100 WAHL$=INKEY$:IF WAHL$="" THEN 2100
2110 IF VAL(WAHL$) <1 THEN GOTO 2100
2120 IF VAL(WAHL$) >9 THEN GOTO 2100
2130 IF VAL(WAHL$) = 9 THEN RETURN
2140 FELDNR = VAL(WAHL$):GOSUB 2200:RETURN
2150 GOTO 2100
2200 XP=2:YP=18:ON FELDNR GOTO 2210,2210,2210,2240
,2240,2210,2270,2300
2210 ML=10:LOCATE XP,YP:PRINT STRING$(ML,"."):GOSU
B 1510
2220 IF ER=0 THEN SUCH$=AN$:GOTO 2330
2230 IF ER=1 THEN GOTO 2210
2240 ML=4:LOCATE XP,YP:PRINT STRING$(ML,"."):GOSUB
 1510
2250 IF ER=0 THEN SUCH$=AN$:GOTO 2330
2260 IF ER=1 THEN GOTO 2240
2270 ML=15:LOCATE XP,YP:PRINT STRING$(ML,"."):GOSU
B 1510
2280 IF ER=0 THEN SUCH$=AN$:GOTO 2330
2290 IF ER=1 THEN GOTO 2290
2300 ML=20:LOCATE XP,YP:PRINT STRING$(ML,"."):GOSU
B 1510
2310 IF ER=0 THEN SUCH$=AN$:GOTO 2330
2320 IF ER=1 THEN GOTO 2300
2330 SUCHLANG=LEN(SUCH$)
2340 FOR SCAN = 0 TO LFD
2350 SCAN$=LEFT$(ADR$(SCAN,FELDNR),SUCHLANG)
2360 IF UPPER$(SUCH$)=UPPER$(SCAN$) THEN GOTO 2400
2370 NEXT SCAN
2380 PRINT:PRINT"NICHT GEFUNDEN [TASTE]":CALL &BB0
6:GOTO 2000
2400 CLS
2410 PRINT"GEFUNDEN : ADRESSE NUMMER ";SCAN:PRINT
2420 PRINT" 1 : NAME    :";ADR$(SCAN,1)
2430 PRINT" 2 : VORNAME:";ADR$(SCAN,2)
2440 PRINT" 3 : STRASSE:";ADR$(SCAN,3)
2450 PRINT" 4 : HSNR.   :";ADR$(SCAN,4)
2460 PRINT" 5 : PLZ.    :";ADR$(SCAN,5)
2470 PRINT" 6 : WOHNORT:";ADR$(SCAN,6)
2480 PRINT" 7 : TEL-NR.:";ADR$(SCAN,7)
2490 PRINT" 8 : BEMERK.:";ADR$(SCAN,8)
2500 PRINT:PRINT"[W] EITER   [A] ENDERN    [M] ENUE
"
2510 ABF$=INKEY$:IF ABF$="" THEN 2510
2520 IF UPPER$(ABF$)="W" THEN 2370
2530 IF UPPER$(ABF$)="A" THEN 2600
2540 IF UPPER$(ABF$)="M" THEN RETURN
2550 GOTO 2510
2600 REM HIERHIN KOMMT MODUL [AENDERN]
```

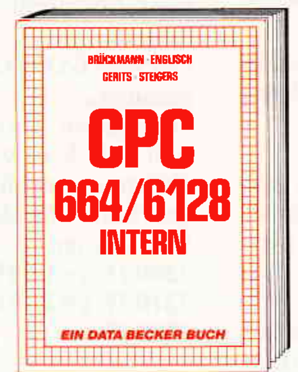
DATA BECKER's gr



Das sollte Ihr erstes Buch zum CPC 6128 sein! CPC 6128 für Einsteiger ist eine sehr leicht verständliche Einführung in Handhabung und Einsatz des CPC 6128, die keinerlei Vorkenntnisse voraussetzt. Dazu eine Einführung in BASIC, wobei Sie eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Unentbehrlich für jeden CPC 6128 Einsteiger!
CPC 6128 für Einsteiger
215 Seiten, DM 29,–



Wollen Sie in BASIC wie ein Profi programmieren? Dieses Buch macht es Ihnen leicht. Themenbereiche: Variablen, Zahlensysteme, Bits und Bytes, Tokens, Stringbearbeitung, Sortierung, Laufschrift, selbstdefinierte Zeichen, Windows, Rundungen, Fehlerbearbeitung, Kopierschutz, Grafiken, Joystick, Soundprogrammierung, relative Dateien u.v.m. Viele Beispielprogramme finden Sie in den entsprechenden Kapiteln.
Das große BASIC-Buch zum 6128
ca. 300 Seiten, DM 39,–



Ein Muß für jeden, der sich professionell mit dem CPC 6128 oder dem CPC 664 beschäftigt. Einführung in das System, den Prozessor, das Gate Array, den Video-Controller, den Schnittstellenbaustein 8255, den Soundchip, die Schnittstellen. Mit Disassembler und ausführlichen Kommentaren zu den Routinen von Interpreter und Betriebssystem. Ein Superbuch, wie alle Titel der INTERN-Reihe!
CPC 664/6128 Intern
456 Seiten, DM 69,–



Von den Grundlagen der Maschinenspracheprogrammierung über die Arbeitsweise des Z80 Prozessors und einer genauen Beschreibung seiner Befehle bis zur Benutzung von Systemroutinen ist alles ausführlich und mit vielen Beispielen erklärt. Im Buch enthalten sind Assembler, Disassembler, Einzelschritt-Simulator und Monitor als komplette Anwenderprogramme. So wird der Einstieg in die Maschinensprache leichtgemacht.
Das Maschinensprachebuch zum CPC
320 Seiten, DM 39,–



Eine Sprache, die immer beliebter wird ist LOGO. LOGO ist einfach zu erlernen, aber vielseitig in der Programmierung. Das Buch befaßt sich unter anderem mit folgenden Themen: Rechnen mit LOGO, Grafikprogrammierung, Wörter- und Listenverarbeitung, Prozeduren, Rekursionen, Sortierroutinen, Maskengenerator, Datenstrukturen und Künstliche Intelligenz.
Das große LOGO-Buch zum CPC
ca. 300 Seiten, DM 39,–

Rund um den CPC 464 viele Anregungen und wichtige Hilfen! Von Hardwareaufbau, Betriebssystem, BASIC-Tokens, Anwendungen der Windowtechnik und sehr vielen interessanten Programmen bis zu einer umfangreichen Dateiverwaltung, Soundeditor, komfortablem Zeichengenerator und kompletten Listings spannender Spiele bietet dieses Buch eine Fülle von Möglichkeiten. Diese Tips kommen von den DATA BECKER Spezialisten!
CPC 464 Tips & Tricks
271 Seiten, DM 39,–



Der 2. Band CPC Tips & Tricks ist für alle CPC Besitzer interessant. Ob sie nun einen 464, 664 oder 6128 besitzen! Aus dem Inhalt: Menuegenerator, Maskengenerator, BASIC-Befehlserweiterungen, Programmierhilfen wie Dump, BASIC-Zeile von BASIC aus erzeugen, wichtige Systemroutinen und deren Nutzung, Beschleunigung von Programmen u.v.m. Wer noch mehr über seinen CPC wissen will, der kommt an diesem Buch nicht vorbei!
CPC Tips & Tricks Band II
250 Seiten, DM 39,–



Endlich CP/M beherrschen! Von grundsätzlichen Erklärungen zu Speicherung von Zahlen, Schreibschutz oder ASCII, Schnittstellen und Anwendung von CP/M-Hilfsprogrammen. Für Fortgeschrittene: Fremde Diskettenformate lesen, Erstellen von Submit-Dateien u.v.m. Dieses Buch berücksichtigt die Versionen CP/M 2.2 und 3.0 für Schneider 464, 664 und 6128.
Das CP/M-Trainingsbuch zum CPC
260 Seiten, DM 49,–



Grundlagen und Struktur von Pascal, Anweisungen, Abweichungen vom Standard-Pascal, Prozeduren/Funktionen, Datentypen, Rekursionen und Compileroperationen sowie Grafik unter PC-DOS/MS-DOS sind die Themen, die in diesem Trainingsbuch umfassend abgehandelt werden. Übungsaufgaben und Beispiele helfen, den Stoff zu erfassen und das Gelernte zu vertiefen.
Das Trainingsbuch zu TURBO PASCAL
269 Seiten, DM 39,–



Eine beispielslose Sammlung von Tips und Tricks, mit denen Sie alle Vorzüge von TURBO PASCAL erfolgreich nutzen können. Natürlich mit vielen Anwendungen und konkreten Programmierhilfen für den optimalen Einsatz dieser erstaunlich vielseitigen Programmiersprache. Ein gelungenes Buch, das reichlich Anregungen vermittelt und damit zu einer wirklichen Fundgrube für jeden Anwender wird.
TURBO PASCAL Tips & Tricks
243 Seiten, DM 49,–



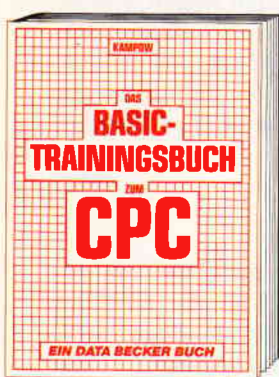
Multiplan ist eines der erfolgreichsten Kalkulationsprogramme! Um die vielen Vorteile eines solchen Programmpaketes nutzen zu können, bedarf es allerdings einer guten Einführung: Das Trainingsbuch ist dazu der optimale Weg. Sicheres Arbeiten und auch die Nutzung des umfangreichen Befehlssatzes für kommerzielle Anwendungen sind damit problemlos möglich!
Das Trainigsbuch zu MULTIPLAN
250 Seiten, DM 49,–



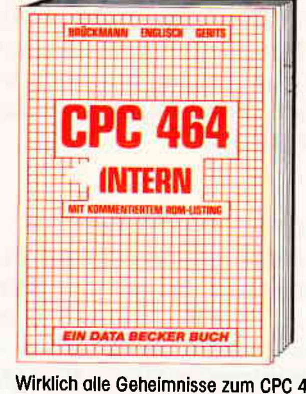
Eine ausführliche und leichtverständliche Einführung in den Umgang mit Datenbanken bietet das Trainingsbuch zu dBASE II. Aus dem Inhalt: Eröffnung und Struktur einer Datenbank in dBASE II, Umgang mit Zahlen in Datenbanken, Daten suchen und löschen, Datenbanken kombinieren, Schleifen, Memoryvariablen, Fehlersuche, Menüs, mit vielen prakischen Hinweisen.
Das Trainingsbuch zu dBASE II
322 Seiten, DM 49,–

Aus Deutschlands größtem Verlag für Computerbücher

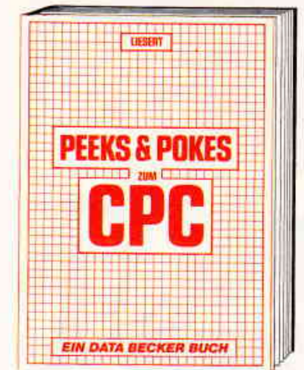
...ße CPC Bibliothek



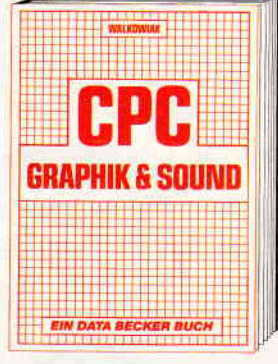
CPC 464 BASIC? Kein Problem! Mit diesem Trainingsbuch lernen Sie von Grund auf nicht nur die einzelnen Befehle und ihre Anwendungen, sondern auch einen richtig sauberen Programmierstil. Von der Problemanalyse über den Datenflußplan bis zum fertigen Programm. Dazu viele Übungsaufgaben mit Lösungen und zahlreichen Beispielen. Schlichtweg unentbehrlich!
Das BASIC-Trainingsbuch zum CPC
285 Seiten, DM 39,–



Wirklich alle Geheimnisse zum CPC 464 lüftet dieses Standardwerk: Neben dem kommentierten ROM-Listing enthält es Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinenutzung, Character-Generator, u.v.m. Für den fortgeschrittenen BASIC-Programmierer unentbehrlich, für den Assembler-Programmierer ein absolutes Muß!
CPC 464 Intern mit kommentiertem ROM-Listing, 548 Seiten, DM 69,–



Wer PEEKS und POKES zum CPC 464 kennen und anwenden will, der findet hier umfassende Information! Sie reicht vom Adreßbereich des Prozessors über Betriebssystem und Interpreter bis hin zur Einführung in die Maschinensprache. Dazu Programmierhilfen, Routinen sowie reichlich Material zu den Themen Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC und RAM-Pages!
Peeks & Pokes zum CPC
180 Seiten, DM 29,–



Nutzen Sie die außergewöhnlichen Grafik- und Soundmöglichkeiten des CPC 464! Natürlich mit vielen interessanten Beispielen und Programmen: Grafikgrundlagen, Sprites, Shapes, Strings, mehrfarbige Darstellungen, Koordinationstransformation, Verschiebungen, Drehungen, Rotation, 3-D-Funktionsplotter, CAD, Synthesizer, Miniorgel, Hüllkurven u.v.m. Dieses Buch wird Sie begeistern!
CPC 464 Grafik & Sound
220 Seiten, DM 39,–



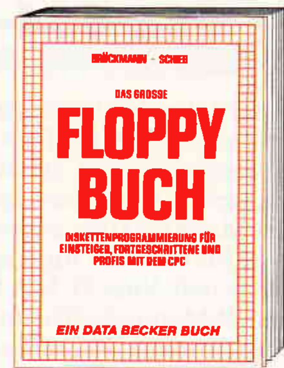
DFÜ für Jedermann mit dem CPC bietet eine ausführliche und verständliche Einführung in das Gebiet der Datenfernübertragung: was ist DFÜ, BTX, DATEX, Mailbox. Alles über Modems und Koppler. Begriffserklärung: Originate, Answer, Half-Duplex usw. Eine serielle Schnittstelle am CPC, RS 232/V.24 simuliert, Mailboxsoftware – selbstgestrickt, Postbestimmungen u.v.m. Steigen Sie mit diesem Buch in die Welt der Datennetze und Datenfernübertragung ein. **DFÜ für Jedermann zum CPC**
303 Seiten, DM 40,–



CAD auf dem CPC. Mit dieser Einführung in Computer Aided Design erlernen auch Sie die Programmierung von komplexen 3-D-Grafiken. Aus dem Inhalt: Punkt, Linie, Rechteck, Kreis, Bogen, 3-D-Körper wie Quader/Würfel, Pyramide, Prisma, Zylinder usw., Draht- und Volumenmodelle, Drehen und Spiegeln von Figuren, Explosionszeichnungen, Platinenlayouts und der Clou: Tips zum Aufbau eines eigenen CAD-Systems!
Einführung in CAD zum CPC
ca. 300 Seiten, DM 49,–



Speziell für den Hobbyelektroniker, der mehr aus seinem CPC machen möchte! Von nützlichen Tips zur Platinenherstellung über Adreßdecodierung, Adapterkarten und Interfaces bis zu EPROM-Programmierboard und -Programmiernetzteil oder Motorsteuerung für Gleich- und Schrittschaltmotoren werden machbare Erweiterungen ausführlich und praxisnah beschrieben. Am besten gleich anfangen!
CPC Hardware-Erweiterungen
445 Seiten, DM 49,–



Alles über Floppyprogrammierung vom Einsteiger bis zum Profi in der jetzt komplett überarbeiteten und erweiterten Neuauflage. Natürlich mit ausführlichen Kommentaren zu allen ROM-Routinen, einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Monitor und einem ausgesprochen nützlichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen!
DAS große Floppy-Buch zum CPC
422 Seiten, DM 49,–



Sie verarbeiten Ihre Texte mit WORDSTAR? Dann werden Sie mit den Tips & Tricks dieses Buches zum Profi. Viele Arbeiten lassen sich effektiver und schneller erledigen. Lassen Sie sich von einem Spezialisten den Weg zur optimalen Ausnutzung aller Stärken von WORDSTAR zeigen, denn oft bleiben viele Anwendungsmöglichkeiten in der täglichen Routine ungenutzt.
WORDSTAR Tips & Tricks
224 Seiten, DM 49,–

Dieses Buch führt Sie Schritt für Schritt in die Benutzung des Joyce ein. Diese Einführung geht von der Installation der Geräte über eine Einleitung in LogoScript bis hin zur Programmerstellung in den Programmiersprachen BASIC und LOGO. Auch die Benutzung des Betriebssystems CP/M 3.0 wird leicht verständlich beschrieben. Der ideale Einstieg mit dem Joyce!
Joyce für Einsteiger
248 Seiten, DM 29,–



Dieses Buch beschreibt umfassend die Benutzung der Textverarbeitung Loco-Script, das Betriebssystem CP/M und die Anwendung von Programmen wie Multiplan, WordStar und dBase, wichtige Mallard-BASIC-Routinen wie Menü- und Maskengenerator, Sortierverfahren und rekursive Grafikprogrammierung in LOGO!
Das große Joyce-Buch
ca. 350 Seiten, DM 59,–

Und wo informieren sich CPC-Anwender über News & Trends, neue Software, neue Computer und aktuelle Tips & Tricks? In der

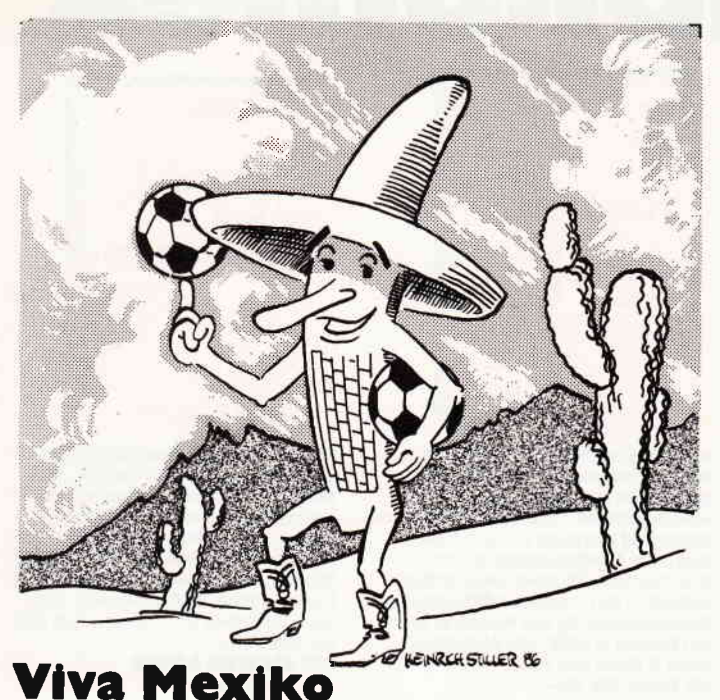
DATA WELT 6/86

Wo denn sonst!
DATA WELT 6/86
ab 20. Mai
am Kiosk.

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010



Viva Mexiko für 464-664-6128

Alle vier Jahre blickt die ganze Welt auf ein massenbewegendes Ereignis: die Fußball-Weltmeisterschaft. Selbst diejenigen, die nicht als ausgesprochene Fußball-Fans gelten, werden von diesem sportlichen Wettkampf der weltbesten Mannschaften angezogen.
1986 ist Mexiko das Austragungsland der Fußball-Weltmeisterschaft. Vom 31. Mai bis zum 29. Juni kämpfen insgesamt 24 Mannschaften um den heißbegehrten Titel und die Siegerprämie. Rechtzeitig und exclusiv für alle CPC-Leser, bringen wir daher ein WM-Programm, mit dem alle anfallenden Spieldaten erfaßt und statistisch aufbereitet werden können.
Das WM-Programm ist leicht zu bedienen und beinhaltet bereits alle teilnehmenden Fußballmannschaften sowie die entsprechenden Spielpaarungen, soweit diese bereits feststehen.

Hinweis:
In Mexiko wird erstmals nach einem neuen Austragungsmodus gespielt: In sechs Gruppen zu je vier Mannschaften, von denen dann die zwei besten jeder Gruppe das Achtel-Finale erreichen.
Erstmals gibt es auch für die Tabellen-Dritten eine Chance, ins Achtel-Finale zu kommen, allerdings jeweils nur die vier Besten von Rang 3. Die erzielten Punkte, die Tordifferenz und die Zahl der geschossenen Tore geben hier den Ausschlag. Sollte sich dabei punktemäßig sowie in der Tordifferenz und in der Anzahl der geschossenen Tore eine Gleichheit ergeben, entscheidet für ein Aufsteigen das Los.
Erst nach den Finalrundenspielen kann über ein Weiterkommen der Tabellen-Dritten entschieden werden, damit auch über eine Einreihung in das Achtel-Finale. Es können also nicht zwei Mannschaften aus der gleichen Gruppe im Achtel-Finale, auch nicht im Viertel- oder Halbfinale, aufeinander treffen.
Nach den Achtel-Finalspielen fallen alle weiteren Entscheidungen im K.O.-System. Die Mannschaft, die verliert, scheidet sofort aus. Dadurch sollen langweilige Spiele, wie es sie zum Beispiel bei der WM in Spanien gegeben hat, vermieden werden. (Wer erinnert sich nicht an das einschläfernde Spiel Österreich - Deutschland?)
Sollten die Spiele, die im K.O.-System ausgetragen werden, nach der normalen Spielzeit unentschieden stehen, gibt es eine Verlängerung von 2 x 15 Minuten. Ist auch dann noch keine Entscheidung gefallen, folgt ein Elfmeter-Schießen.
Bleibt zum Schluß nur noch der Wunsch, daß in einem sportlich fairen Turnier die beste Mannschaft als Sieger hervorgeht.

(SR)

BEDIENUNGSANLEITUNG:
Nach dem Eintippen wird das Programm mit 'SAVE "WM"' abgespeichert. Beim ersten Start mit 'RUN' wird eine 2k-Datei "ERGEBNIS.DAT" erzeugt, worin die bekannten Spielpaarungen und Ergebnisse abgelegt werden.
Im Hauptmenue gibt es fünf Optionen:
- Mit <E>ingabe kommt man in das Eingabe-Untermenue.
- Mit<A>usgabe kommt man in das Ausgabe-Untermenue.
- Mit <Q>uit kann man das Programm verlassen. Allerdings wird vorher noch gefragt, ob man es wirklich verlassen möchte.
- Mit <B>efehle kann man eine Kurzbeschreibung der einzelnen Befehle abrufen.
- Mit<TAB> kann man einen Bildschirm-Ausdruck auf einem Drucker erzeugen. Auch hier wird vorher gefragt, ob der Drucker bereit ist. Falls nicht, springt das Programm ins aufrufende Menue zurück.

Im Eingabemenue gibt es fünf Optionen:
- Mit <V>orrunden kann man die Ergebnisse der Vorrundenspiele 1 - 36 eingeben, wobei das Programm erklärt, was zu tun ist.
- Mit <H>auptrunde kann man die Ergebnisse der Hauptrundenspiele eingeben. Dies ist aber nur möglich, wenn beide Mannschaften eines Spieles bekannt sind. Der Sieger wird dann automatisch in die nächste Runde übernommen.
- Mit <E>ingabe von Mannsch. ist es möglich, Mannschaften, die nicht automatisch übertragen werden, einzugeben.
  Hierzu können mehrere Ursachen führen:
  - In den Spielen 37 - 40 sind jeweils die zweiten Spielpartner unbekannt. Es werden Gruppendritte der Vorrunde sein.
  - Falls in einer Vorrundengruppe sowohl Punkt- als auch Torgleichheit vorkommt, ist es theoretisch möglich, daß eine falsche Mannschaft in die Hauptrunde übertragen wird (Auslosung).

  In solchen Fällen muß diese Option benutzt werden. Erst wird nach der Spielnummer gefragt. Danach muß eingegeben werden, ob es sich um die erste oder zweite Mannschaft einer Paarung handelt. Zum Schluß wird nach dem Mannschaftsnamen gefragt. Hierbei muß auf die Rechtschreibung, nicht aber auf die Groß- und Kleinschreibung geachtet werden.
- <Q>uit zur Rückkehr zum Hauptmenue.
- <B>efehle wie im Hauptmenue.

Im Ausgabemenue gibt es sechs Optionen:
- Mit <V>orrunde werden die Vorrundenspiele und -ergebnisse ausgegeben.
- Mit<T>abellen werden die Vorrundentabellen errechnet und ausgegeben.
- Mit<H>auptrunde werden die Hauptrundenpaarungen ermittelt und zusammen mit den bekannten Ergebnissen ausgegeben.
- <Q>uit zur Rückkehr zum Hauptmenue.
- <B>efehle wie im Hauptmenue
- <TAB> wie im Hauptmenue (Hardcopy).

Wenn Sie einen CPC 6128 besitzen, so können Sie noch den Ergänzungsteil eintippen und mit 'SAVE' "ERGAENZU",A' abspeichern. Danach laden Sie das Hauptprogramm mit 'LOAD "WM"' ein und laden den Ergänzungsteil mit 'MERGE "ERGAENZU"' dazu. Diese Version speichern Sie mit 'SAVE "WMV6128"' ab. Danach kopieren Sie noch die Programme "BANKMAN.BAS" und "BANKMAN.BIN" von Ihrer CP/M-Systemdiskette mit "FILECOPY" auf Ihre Diskette.
Um das WM-Programm nun zu starten, müssen Sie zweimal 'RUN "WMV6128"' eingeben.

Sonderfunktionen mit Bankmanager
Im Haupt- und Ausgabemenue ist es möglich, durch Drücken der <SPACE>-Taste in das Bankmanager-Untermenue zu gelangen. Dort gibt es drei Optionen:
- Mit <K>opieren kann man den Inhalt eines Bildschirmes in einen anderen kopieren. Der Inhalt des anderen geht verloren.
- Mit<T>auschen kann man zwei Bildschirminhalte vertauschen.
- <Q>uit zur Rückkehr zum Hauptmenue.

Erklärung:
Beim CPC 6128 existiert eine zweite Speicherbank, in die vier Bildschirminhalte passen. Bei den Bankmanagerfunktionen haben diese die Nummern 2 - 5, der sichtbare Bildschirm die Nummer 1.
Will man z.B. die Vorrundentabellen später nochmals betrachten, so erzeugt man sie erst einmal, geht ins Bankmanagermenue und kopiert Bildschirm Nr. 1 nach Nr. 2. Später kann man jederzeit durch Kopieren von Nr. 2 nach Nr. 1 diese Tabellen betrachten, ohne die Zeit zur Berechnung abwarten zu müssen.

(M. Braun)

Hauptlisting

```
1 '           ****************************** [1671]
***********
2 '           *                              [175]
                                   *
3 '           *  (c) 1986 by M.Braun & W.Fu  [2426]
ssangel   *
4 '           *                              [175]
                                   *
5 '           ****************************** [1671]
***********
6 '                                          [117]
8 MEMORY &9FFF:FOR adr=&A000 TO &A007:READ   [4640]
 wert$:POKE adr,VAL ("&"+wert$):NEXT adr:D
ATA cd,60,bb,32,07,a0,c9,00
10 MODE 2                                    [513]
11 DIM e%(51,1),s%(51,1),t%(23,4)            [1599]
12 DEFINT a-z                                [553]
20 INK 1,0: INK 0,13: BORDER 13              [1384]
21 PRINT"Kasetten oder Diskettenbetrieb [K   [3783]
/D]"
22 devi$=INKEY$:IF devi$="" THEN 22          [1104]
23 IF UPPER$(devi$)="K" THEN cas=111:GOTO    [1839]
30
24 IF UPPER$(devi$)="D" THEN cas=0:|DISC:G   [3877]
OTO 30
25 GOTO 22                                   [396]
30 REM                                       [272]
90 '                                         [117]
100 'Daten laden                             [1528]
110 '                                        [117]
120 PRINT"Besteht bereits eine Datei [J/N]   [3957]
"
121 abf$=INKEY$:IF abf$="" THEN 121          [2574]
122 IF UPPER$(abf$)="J" THEN 130             [1601]
123 IF UPPER$(abf$)="N" THEN 210             [1506]
124 GOTO 121                                 [432]
129 IF  cas THEN PRINT"Datenkassette einle   [7507]
gen, zurueckspulen und Taste druecken":CAL
L &BB06
130 OPENIN "ergebnis.dat"                    [2229]
140 ON ERROR GOTO 0                          [1381]
150 FOR i=0 TO 51: INPUT #9,e%(i,0),e%(i,1   [2286]
): NEXT i
160 FOR i=0 TO 51: INPUT #9,s%(i,0),s%(i,1   [1878]
): NEXT i
170 CLOSEIN                                  [752]
190 GOTO 310                                 [510]
200 '                                        [117]
210 'Datei erzeugen                          [1740]
220 IF  cas THEN PRINT"Datenkassette einle   [5065]
gen und Taste druecken":CALL &BB06
230 OPENOUT "ergebnis.dat"                   [1446]
240 FOR i=0 TO 103: PRINT #9,-1: NEXT i      [994]
250 RESTORE 2010: FOR i=0 TO 35: FOR k=0 T   [1715]
O 1
260 READ t: PRINT #9,t: NEXT k: NEXT i       [1125]
270 RESTORE 2450: FOR i=72 TO 103: READ t:   [3661]
 PRINT#9,t: NEXT i
280 FOR i=0 TO 31: PRINT #9,-1: NEXT i       [1126]
290 CLOSEOUT                                 [902]
300 GOTO 129                                 [424]
310 '                                        [117]
320 'Menue                                   [1057]
330 '                                        [117]
340 MODE 2: DIM n$(23)                       [1658]
350 '                                        [117]
360 'Einsprung                               [1444]
370 '                                        [117]
380 DEF FNf$(x)=RIGHT$(STR$(x),LEN(STR$(x)   [2644]
)-1)
390 RESTORE 1950: FOR i=0 TO 23: READ n$(i   [3125]
): NEXT i
400 WINDOW #2,1,80,2,25                      [1246]
410 WINDOW #1,1,80,1,1                       [995]
420 CLS #1                                   [373]
430 PRINT#1,"<E>ingabe   <A>usgabe   <Q>ui   [4257]
t   <B>efehle   <TAB>"
440 x$="EABQ"+CHR$(9): GOSUB 1850            [2601]
450 IF a$="E" THEN GOSUB 510: GOTO 420       [2122]
460 IF a$="A" THEN GOSUB 720: GOTO 420       [1633]
470 IF a$="B" THEN GOSUB 3210: GOTO 420      [1621]
490 IF a$=CHR$(9) THEN GOSUB 2330: GOTO 42   [1264]
0
500 IF a$="Q" THEN GOTO 1750                 [1114]
510 '                                        [117]
520 'Eingabe                                 [925]
530 '                                        [117]
540 CLS #1                                   [373]
550 PRINT#1,"Inp.: <V>orrunde   <H>auptrun   [8048]
de   <E>ingabe von Mannsch.   <Q>uit   <B>
efehle"
560 b$=" ": x$="VHEQB": GOSUB 1850           [2262]
570 IF a$="V" THEN b$="V": GOSUB 850: GOSU   [3198]
B 620: RETURN
580 IF a$="H" THEN b$="H":GOSUB 3020:  GOS   [4572]
UB 960: GOSUB 620: RETURN
590 IF a$="E" THEN GOSUB 3070: GOTO 540      [2069]
600 IF a$="B" THEN GOSUB 3210:  GOTO 550     [1643]
610 IF a$="Q" THEN RETURN                    [1275]
620 CLS #1: INPUT#1,"Spielnummer ";a$:a=IN   [2895]
T(VAL(a$)-1)
630 IF a<0 OR a>51 THEN RETURN               [1137]
640 IF s(a,0)<0 OR s(a,1)<0 THEN RETURN      [3314]
650 CLS #1: PRINT#1,a+1;": ";n$(s(a,0));:    [3646]
INPUT #1," --> ";a$:b=INT(VAL(a$))
660 CLS #1: PRINT#1,a+1;": ";n$(s(a,1));:    [4692]
INPUT #1," --> ";a$:c=INT(VAL(a$))
670 CLS #1: PRINT#1,n$(s(a,0));" : ";n$(s(   [4067]
a,1));"   ";b;":";c;"   ";
680 PRINT#1,"korrekt ?": x$="JN": GOSUB 18   [2680]
50
690 IF a$="N" THEN 620                       [943]
700 e(a,0)=b: e(a,1)=c: IF b$="H" THEN GOS   [2625]
UB 2540: GOSUB 960: GOTO 620
710 GOSUB 850: GOTO 620                      [1338]
720 '                                        [117]
730 'Ausgabe                                 [572]
740 '                                        [117]
750 CLS #1                                   [373]
760 PRINT#1,"<V>orrunde   <T>abellen   <H>   [5889]
auptrunde   <Q>uit   <B>efehle   <TAB>"
770 x$="VHTQB"+CHR$(9): GOSUB 1850           [1377]
780 IF a$="V" THEN GOSUB 850: GOTO 750       [2087]
790 IF a$="H" THEN GOSUB 960: GOTO 750       [1873]
800 IF a$="T" THEN GOSUB 1280: GOTO 750      [2532]
810 IF a$="B" THEN GOSUB 3210: GOTO 750      [1667]
830 IF a$=CHR$(9) THEN GOSUB 2330: GOTO 75   [1414]
0
```

```
840 IF a$="Q" THEN RETURN                       [1275]
850 '                                           [117]
860 'Vorrundenergebnisse                        [1346]
870 '                                           [117]
880 CLS #2: FOR y=0 TO 2: FOR x=0 TO 1          [1592]
890 x1=x*40+1: y1=y*8+3: REM Linke obere E      [3690]
cke
900 LOCATE x1+1,y1: PRINT"Gruppe ";CHR$(65      [2091]
+y*2+x)
910 FOR t=0 TO 5                                [590]
920 LOCATE x1,y1+1+t:z=(y*2+x)*6+t              [1919]
930 GOSUB 1190: REM Paarung ausgeben            [2097]
940 NEXT t: NEXT x: NEXT y                      [1028]
950 RETURN                                      [555]
960 '                                           [117]
970 'Hauptrundenergebnisse                      [2423]
980 '                                           [117]
990 CLS #2: GOSUB 3020                          [649]
1000 x1=1: y1=3                                 [1023]
1010 LOCATE x1+1,y1: PRINT"Achtelfinale"        [2151]
1020 FOR z=36 TO 43: y1=y1+1: LOCATE x1,y1      [2799]
1030 GOSUB 1190: NEXT z                         [1612]
1040 x1=41: y1=3                                [533]
1050 LOCATE x1+1,y1: PRINT"Viertelfinale"       [2527]
1060 FOR z=44 TO 47: y1=y1+1: LOCATE x1,y1      [2849]
1070 GOSUB 1190: NEXT z                         [1612]
1080 x1=41: y1=9                                [559]
1090 LOCATE x1+1,y1: PRINT"Halbfinale"          [3077]
1100 FOR z=48 TO 49: y1=y1+1: LOCATE x1,y1      [2607]
1110 GOSUB 1190: NEXT z                         [1612]
1120 x1=1: y1=15                                [974]
1130 LOCATE x1+1,y1: PRINT"Spiel um Platz       [3475]
drei"
1140 z=50: y1=y1+2: LOCATE x1,y1: GOSUB 11      [2432]
90
1150 x1=41: y1=15                               [633]
1160 LOCATE x1+1,y1: PRINT"Endspiel"            [2109]
1170 z=51: y1=y1+2: LOCATE x1,y1: GOSUB 11      [2098]
90
1180 RETURN                                     [555]
1190 '                                          [117]
1200 'Ausgabe einer Paarung                     [1400]
1210 '                                          [117]
1220 PRINT TAB(1+x1)RIGHT$(STR$(z+1),2);"       [1875]
";
1230 IF s%(z,0)>-1 THEN PRINT n$(s%(z,0));      [1918]
1240 PRINT TAB(17+x1)"- ";: IF s%(z,1)>-1       [2816]
THEN PRINT n$(s%(z,1));
1250 IF e%(z,0)<0 THEN 1270                     [1132]
1260 PRINT TAB(32+x1);USING"\\";FNf$(INT(e      [4473]
%(z,0)));":";FNf$(INT(e%(z,1)))
1270 RETURN                                     [555]
1280 '                                          [117]
1290 'Vorrundentabellen                         [1640]
1300 '                                          [117]
1310 CLS #2: fl=0:  GOTO 1330                   [814]
1320 fl=1: REM ohne Ausgabe der Tabelle         [1484]
1330 FOR i=0 TO 23                              [508]
1340 t%(i,0)=0: t%(i,1)=0: t%(i,2)=0: t%(i      [2952]
,3)=0: t%(i,4)=i: NEXT i
1350 FOR z=0 TO 35                              [968]
1360 IF e%(z,0)=-1 THEN 1450                    [865]
1370 IF e%(z,0)<=e%(z,1) THEN 1390              [850]
1380 t%(s(z,0),0)=t%(s(z,0),0)+2: t%(s(z,1      [3185]
),1)=t%(s(z,1),1)+2: GOTO 1430
1390 IF e%(z,0)>=e%(z,1) THEN 1410              [1004]
1400 t%(s(z,1),0)=t%(s(z,1),0)+2: t%(s(z,0      [4890]
),1)=t%(s(z,0),1)+2: GOTO 1430
1410 t%(s(z,0),0)=t%(s(z,0),0)+1: t%(s(z,1      [3165]
),0)=t%(s(z,1),0)+1
1420 t%(s(z,0),1)=t%(s(z,0),1)+1: t%(s(z,1      [3357]
),1)=t%(s(z,1),1)+1
1430 t%(s(z,0),2)=t%(s(z,0),2)+e%(z,0): t%      [4673]
(s(z,1),2)=t%(s(z,1),2)+e%(z,1)
1440 t%(s(z,0),3)=t%(s(z,0),3)+e%(z,1): t%      [3544]
(s(z,1),3)=t%(s(z,1),3)+e%(z,0)
1450 NEXT z                                     [354]
1460 FOR i=0 TO 23 STEP 4                       [1200]
1470 FOR k=i TO i+2: FOR l=i+2 TO k STEP-1      [2448]
1480 x=t%(l,4): y=t%(l+1,4)                     [2136]
1490 IF t%(x,0)>t%(y,0) THEN 1590               [1631]
1500 IF t%(x,0)<t%(y,0) THEN 1570               [1089]
1510 IF t%(x,1)<t%(y,1) THEN 1590               [1726]
1520 IF t%(x,1)>t%(y,1) THEN 1570               [2554]
1530 d=t%(x,2)-t%(x,3): e=t%(y,2)-t%(y,3)       [3051]
1540 IF d>e THEN 1590                           [1228]
```

Sie wollen uns ein Programm zusenden? Folgende Regeln sollten Sie beachten:

1) Schicken Sie Ihre Programme auf Kassette oder Diskette. Dabei mehrmals hintereinander abspeichern.

2) Legen Sie eine ausführliche Programmbeschreibung bei, aus der eindeutig der Sinn und Zweck des Programmes hervorgeht. Eine Variablenliste ist sehr wichtig, in ihr sollten alle im Programm definierten Variablen mit ihrer Verwendung aufgeführt sein.
Beispiel: n$ = Nachname
g = glücklicher usw.

3) Wenn Sie gleichzeitig Besitzer eines Druckers sind, schicken Sie auf jeden Fall ein Programmlisting mit.

4) Vergessen Sie Ihren Absender nicht! Vollständige Anschrift, wenn möglich auch die Telefonnummer, ersparen unnötige Verzögerungen bei eventuellen Nachfragen unsererseits.

5) Behalten Sie auf jeden Fall eine Kopie Ihres Programmes, unvorhersehbare Umstände, wie z.B. Bandriß der Kassette, werden somit nicht zur Katastrophe.

6) Wenn Sie sogar mehrere Programme geschrieben haben und diese veröffentlichen wollen, nehmen Sie bitte für jedes Programm eine separate Kassette (mehrmals abspeichern!).

Damit erleichtern Sie unsere Arbeit wesentlich und eine schnellere Bearbeitung wird möglich.

7) Versehen Sie das Programm mit den sehr wichtigen REM-Statements. Das fördert nicht nur die Übersicht für uns und unsere Leser, sondern unterstützt zusätzlich das strukturierte Programmieren. Merke: Auch erfahrene Programmierer verwenden REM-Zeilen, das Programm sollte schließlich jeder verstehen und anwenden können.

8) Vermeiden Sie möglichst Variablennamen, wo leicht l und i oder 0 und o verwechselt werden können. Variablen sollten grundsätzlich in Kleinbuchstaben geschrieben werden und möglichst verständlich sein. Die Variable butter sagt mehr aus, als nur bu.

9) Bieten Sie das Programm niemals verschiedenen Verlagen gleichzeitig an! Mit der Veröffentlichung und dem daraus resultierenden Honorar, gehen die Urheberrechte an den Verlag über! Sollte Ihr Programm dann noch in einer anderen Zeitschrift abgedruckt werden, liegt ein Verstoß gegen das Wettbewerbsrecht vor.

10) Jedes bei uns ordnungsgemäß eingegangene Programm wird ausführlich begutachtet und getestet. Sie erhalten von uns dann umgehend Bescheid, ob Ihr Programm veröffentlicht wird oder nicht. Wenn Sie das Programm nach unserer Begutachtung zurückhaben wollen, legen Sie bitte der Einsendung einen frankierten und adressierten Rückumschlag bei. Die Kosten einer eventuellen Rücksendung sind in jedem Fall vom Einsender zu tragen.

Falls wir uns für den Abdruck Ihres Programmes entschieden haben, setzen wir uns unverzüglich mit Ihnen in Verbindung und besprechen Ihr verdientes Honorar!
Übrigens wäre es nett, wenn Sie uns auch ein paar persönliche Daten schreiben würden (z.B. Ihr Alter, Tätigkeit, wie und wann kam die Idee zu dem Programm, was alles machen Sie mit dem CPC usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre „Schneider CPC international" auf Ihre Beiträge.

Einsendungen bitte an: DMV - Daten & Medien Verlagsges.mbH, Fuldaer Straße 6, 3440 Eschwege

```
1550 IF d<e THEN 1570                          [1127]
1560 IF t%(x,2)>=t%(y,2) THEN 1590             [993]
1570 'Swap                                     [217]
1580 t%(l,4)=y: t%(l+1,4)=x                    [1734]
1590 NEXT l: NEXT k: NEXT i                    [1004]
1600 IF fl=1 THEN RETURN                       [674]
1610 '                                         [117]
1620 'Tabellen ausgeben                        [1475]
1630 '                                         [117]
1640 CLS #2: FOR y=0 TO 2: FOR x=0 TO 1        [1592]
1650 xl=x*40+1: yl=y*8+3: REM Linke obere      [2632]
Ecke
1660 LOCATE xl,yl: PRINT"Gruppe ";CHR$(65+     [3583]
y*2+x)
1670 LOCATE xl+16,yl+1: PRINT"Punkte  Diff     [2642]
. Tore"
1680 FOR t=0 TO 3: z=y*8+x*4+t: z=t%(z,4)      [2257]
1690 LOCATE xl,yl+2+t: PRINT STR$(INT(t+1)     [2679]
);". ";
1700 PRINT n$(z);TAB(xl+16);USING"\\";FNf$     [3466]
(t%(z,0));": ";FNf$(t%(z,1));
1710 PRINT TAB(xl+24);USING"\ \";STR$(t%(z     [3073]
,2)-t%(z,3));
1720 PRINT TAB(xl+29);USING"\\";FNf$(t%(z,     [3457]
2));": ";FNf$(t%(z,3))
1730 NEXT t: NEXT x: NEXT y                    [1028]
1740 RETURN                                    [555]
1750 '                                         [117]
1760 'Beenden                                  [1019]
1770 '                                         [117]
1780 CLS #1: PRINT#1,"Programm verlassen.      [4968]
Sind Sie sicher ?"
1790 x$="JN": GOSUB 1850: IF a$<>"J" THEN      [2743]
420
1795 IF cas THEN PRINT"Datenkassette einl      [7507]
egen, zurueckspulen und Taste druecken":CA
LL &BB06
1800 OPENOUT "ergebnis.dat"                    [1446]
1810 FOR i=0 TO 51: PRINT #9,e%(i,0),e%(i,     [2380]
1): NEXT i
1820 FOR i=0 TO 51: PRINT #9,s%(i,0),s%(i,     [2007]
1): NEXT i
1830 CLOSEOUT: CLS                             [1521]
1840 END                                       [110]
1850 '                                         [117]
1860 'Tastaturabfrage                          [1058]
1870 '                                         [117]
1880 CALL &BB03                                [396]
1890 a$=UPPER$(INKEY$): IF a$="" THEN 1890     [1878]
1900 IF INSTR(x$,a$)<>0 THEN SOUND 1,80,10     [1898]
: RETURN
1910 GOTO 1890                                 [361]
1920 '                                         [117]
1930 'Paarungen der Vorrunde                   [1573]
1940 '                                         [117]
1950 DATA "Italien","Bulgarien","Argentini     [2814]
en","Suedkorea"
1960 DATA "Mexiko","Belgien","Paraguay","I     [3307]
rak"
1970 DATA "Frankreich","Kanada","UdSSR","U     [2393]
ngarn"
1980 DATA "Brasilien","Spanien","Nordirlan     [4226]
d","Algerien"
1990 DATA "Deutschland","Uruguay","Schottl     [4022]
and","Daenemark"
2000 DATA "Polen","Marokko","England","Por     [2919]
tugal"
2010 DATA 0,1,2,3,0,2,1,3,0,3,2,1              [1652]
2020 DATA 4,5,6,7,6,4,5,7,4,7,6,5              [1110]
2030 DATA 8,9,10,11,10,8,11,9,8,11,10,9        [838]
2040 DATA 12,13,14,15,12,15,13,14,12,14,13     [1839]
,15
2050 DATA 16,17,18,19,16,18,17,19,16,19,17     [1778]
,18
2060 DATA 20,21,22,23,22,21,20,23,23,21,20     [1806]
,22
2330 '                                         [117]
2340 'Hardcopy                                 [320]
2350 '                                         [117]
2360 CLS #1: PRINT#1,"Drucker bereit (J/N)     [2221]
?"
2370 x$="JNQ": GOSUB 1850                      [894]
2380 IF a$<>"J" THEN RETURN                    [1001]
2390 FOR k=2 TO 25: a$="": FOR i=1 TO 80       [2107]
2400 LOCATE i,k: CALL &A000: a$=a$+CHR$(PE     [4259]
EK(&A007)): NEXT i: PRINT#8,a$: NEXT k
2410 RETURN                                    [555]
2420 '                                         [117]
2430 'Spiele der Hauptrunde 37-52              [1085]
```

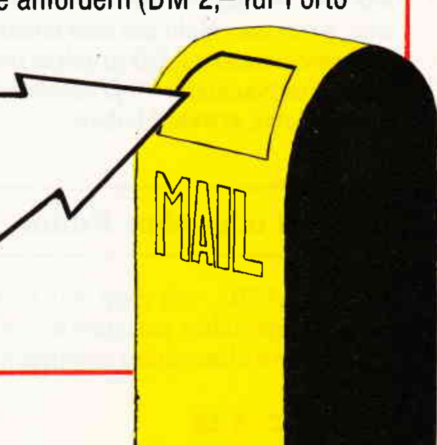



```
2440 '                                          [117]
2450 DATA -102,-400,-103,-400,-101,-400,-1 [2204]
04,-400
2460 DATA -201,-203,-106,-205,-206,-202,-1 [2722]
05,-204
2470 DATA -40,-41,-42,-37,-44,-38,-39,-43  [2227]
2480 DATA -47,-48,-45,-46                  [874]
2490 DATA -349,-350                        [327]
2500 DATA -49,-50                          [484]
2510 '                                     [117]
2520 'Berechnen der Hauptrunde             [1446]
2530 '                                     [117]
2540 GOSUB 1320: REM Tabelle ohne Ausgabe  [3097]
2550 FOR i=36 TO 51                        [1321]
2560 x=s(i,0): GOSUB 2590: s(i,0)=x        [2633]
2570 x=s(i,1): GOSUB 2590: s(i,1)=x        [2080]
2580 NEXT i: RETURN                        [719]
2590 '                                     [117]
2600 'Dekodierung der Spiele der Hauptrund [1406]
e
2610 '                                     [117]
2620 IF x=-400 THEN RETURN: 'Mannschaft un [4338]
bekannt (3.einer Gruppe)
2630 IF x>-300 THEN 2720                   [940]
2640 '                                     [117]
2650 'Verlierer aus Spiel Nr. -3xx         [1938]
2660 '                                     [117]
2670 m=-x-301                              [422]
2680 IF s(m,0)<0 OR s(m,1)<0 THEN RETURN   [3312]
2690 IF e(m,0)<e(m,1) THEN x=s(m,0): RETUR [1938]
N
2700 IF e(m,0)>e(m,1) THEN x=s(m,1)        [505]
2710 RETURN                                [555]
2720 IF x>-200 THEN 2820                   [1127]
2730 '                                     [117]
2740 'Gruppenzweiter aus Gruppe -2xx       [2200]
2750 '                                     [117]
2760 m=-x-201: m=4*m                       [724]
2770 e=0: FOR k=m TO m+3                   [1310]
2780 IF t%(k,0)+t%(k,1)<>6 THEN e=1        [2019]
2790 NEXT k                                [373]
2800 IF e=0 THEN x=t%(m+1,4)               [986]
2810 RETURN                                [555]
2820 IF x>-100 THEN 2920                   [1414]
2830 '                                     [117]
2840 'Gruppensieger aus Gruppe -1xx        [1569]
2850 '                                     [117]
2860 m=-x-101: m=4*m                       [721]
2870 e=0: FOR k=m TO m+3                   [1310]
2880 IF t%(k,0)+t%(k,1)<>6 THEN e=1        [2019]
2890 NEXT k                                [373]
2900 IF e=0 THEN x=t%(m,4)                 [975]
2910 RETURN                                [555]
2920 IF x>=0 THEN RETURN                   [1580]
2930 '                                     [117]
2940 'Sieger aus Spiel Nr. -xx             [1989]
2950 '                                     [117]
2960 m=-x-1                                [415]
2970 IF s(m,0)<0 OR s(m,0)<0 THEN RETURN   [2422]
2980 IF e(m,0)<0 THEN RETURN               [2117]
2990 IF e(m,0)>e(m,1) THEN x=s(m,0): RETUR [982]
N
3000 IF e(m,0)<e(m,1) THEN x=s(m,1): RETUR [1485]
N
3010 RETURN                                [555]
3020 '                                     [117]
3030 'Initialisieren der Hauptrunde noetig [3183]
 ?
3040 '                                     [117]
3050 IF s%(36,0)<0 THEN GOSUB 2540         [1211]
3060 RETURN                                [555]
3070 '                                     [117]
3080 'Eingabe einer Mannschaft, die nicht  [3767]
berechnet wurde
3090 '                                     [117]
3100 CLS #1: INPUT#1,"Spielnummer ";a$:a=I [2895]
NT(VAL(a$)-1)
3110 IF a<36 OR a>51 THEN RETURN           [1732]
3120 CLS #1: PRINT#1,"Die erste oder zweit [4840]
e Mannschaft (1/2/Q)"
3130 x$="12Q": GOSUB 1850                  [1741]
3140 IF a$="Q" THEN RETURN                 [1275]
3150 m=VAL(a$)-1: CLS #1: INPUT#1,"Name "; [2742]
a$: a$=UPPER$(a$)
3160 k=-1: FOR i=0 TO 23: IF a$=UPPER$(n$( [1891]
i)) THEN k=i
3170 NEXT i                                [375]
3180 IF k>-1 THEN s(a,m)=k: GOSUB 960: RET [1902]
URN
3190 PRINT#1,"Fehler   <SPACE>"            [1396]
3200 x$=" ": GOSUB 1850: RETURN            [1305]
3210 '                                     [117]
3220 'Helppage                             [369]
3230 '                                     [117]
3240 CLS #2: LOCATE 1,4                    [1023]
3250 PRINT"<E>ingabe -- <V>orrunde    : Ei [3675]
ngabe der Spiele 1-36": PRINT
3260 PRINT SPC(13)"<H>auptrunde   : Berechn [7486]
ung der bekannten Spielpartner und"
3270 PRINT SPC(29)"Eingabe der Spielergebn [5994]
isse 37-52": PRINT: PRINT SPC(13);
3280 PRINT "<E>ing.v.Mannsch.: Hiermit koe [8585]
nnen Mannschaften eingegeben werden,"
3290 PRINT SPC(32)"die nicht berechnet wer [4526]
den konnten."
3300 PRINT SPC(32)"Auf Gross- Kleinschreib [4417]
ung ist dabei nicht zu"
3310 PRINT SPC(32)"achten.": PRINT: PRINT  [1720]
3320 PRINT "<A>usgabe -- <V>orrunde     : A [5643]
usgabe der Spiele 1-36": PRINT
3330 PRINT SPC(13)"<T>abellen     : Berechn [5561]
ung und Ausgabe Vorrundentabellen"
3340 PRINT                                 [361]

3350 PRINT SPC(13)"<H>auptrunde   : Berechn [6850]
ung und Ausgabe der Spiele 37-52"
3360 PRINT :PRINT                          [743]
3370 PRINT "<TAB> zum Erzeugen eines Bilds [4909]
chirmausdrucks auf dem Drucker."
3380 PRINT :PRINT                          [743]
3390 PRINT "<Q>uit zum Verlassen der Befeh [6550]
lsebene oder des Programms."
3400 RETURN                                [555]
```

Änderung 6128

```
7 ZUSATZPROGRAMM FUER 6128: MIT SAVE "NAME [6051]
",A ABSPEICHERN UND MIT MERGE IN 464 VERSI
ON EINLADEN
50 ON ERROR GOTO 80                        [1546]
60 |SCREENCOPY,0,0                         [1133]
70 GOTO 90                                 [312]
80 RUN "bankman.bas"                       [1656]
180 CLS: FOR i=2 TO 5: |SCREENCOPY,i,1: NE [3090]
XT i
430 PRINT#1,"<E>ingabe   <A>usgabe   <Q>ui [5080]
t   <B>efehle   <SPACE>   <TAB>"
440 x$="EABQ "+CHR$(9): GOSUB 1850         [1165]
480 IF a$=" " THEN GOSUB 2100: GOTO 420    [1495]
760 PRINT#1,"<V>orrunde    <T>abellen   <H> [6248]
auptrunde   <Q>uit   <B>efehle   <SPACE>
 <TAB>"
770 x$="VHTQB "+CHR$(9): GOSUB 1850        [1501]
780 IF a$="V" THEN GOSUB 850: GOTO 750     [2087]
820 IF a$=" " THEN GOSUB 2100: GOTO 750    [1541]
2070 '                                     [117]
2080 'Bankmanager                          [1524]
2090 '                                     [117]
2100 CLS #1: PRINT#1,"Bankmanager: <K>opie [3363]
ren    <T>auschen    <Q>uit"
2110 x$="KTQ": GOSUB 1850                  [492]
2120 IF a$="T" THEN GOSUB 2150: GOTO 2100  [2655]
2130 IF a$="K" THEN GOSUB 2240: GOTO 2100  [1303]
2140 IF a$="Q" THEN RETURN                 [1275]
2150 '                                     [117]
2160 'Screenswap                           [705]
2170 '                                     [117]
2180 CLS #1: PRINT#1,"Tausche Bild Nr. ";  [2419]
2190 x$="12345Q": GOSUB 1850: IF a$="Q" TH [3026]
EN RETURN
2200 i=VAL(a$): PRINT#1,a$;" mit Bild Nr.  [2499]
"
2210 x$="12345Q": GOSUB 1850: IF a$="Q" TH [3026]
EN RETURN
2220 k=VAL(a$): |SCREENSWAP,i,k            [982]
2230 RETURN                                [555]
2240 '                                     [117]
2250 'Screencopy                           [850]
2260 '                                     [117]
2270 CLS #1: PRINT#1,"Kopiere Bild Nr. ";  [2673]
2280 x$="12345Q": GOSUB 1850: IF a$="Q" TH [3026]
EN RETURN
2290 i=VAL(a$): PRINT#1,a$;" nach Bild Nr. [1935]
"
2300 x$="12345Q": GOSUB 1850: IF a$="Q" TH [3026]
EN RETURN
2310 k=VAL(a$): |SCREENCOPY,k,i            [651]
2320 RETURN                                [555]
```


MAIL



CP/M und seine Möglichkeiten

In Teil 3 dieser Serie wurden die Vor- und Nachteile von Compilern gegenüber Interpretern beim Programmieren mit BASIC, PASCAL und FORTRAN besprochen. Ein wichtiger Aspekt war auch das Aufspüren und Verbessern von sogenannten Runtime-Fehlern. Je nach Programmiersprache konnten verschiedene Wege aufgezeigt werden, wobei TURBO PASCAL Pluspunkte durch besonders komfortable Eigenschaften des integrierten Editors beim Bearbeiten der Source-Programme für sich verbuchte. Als Nachteil mußte man allerdings den dadurch verkleinerten Programmspeicherplatz in Kauf nehmen. PASCAL MT+, auf das wir später auch noch näher eingehen wollen, kann zwar mehr Source-Code in einem Rutsch verarbeiten, dafür muß jedoch zusätzlich ein Screen-Editor, wie z.B. WORDSTAR, erworben werden, wenn man nicht mit dem leider nur Zeilen-orientierten ED arbeiten möchte, dessen Nachteile wir ja bereits in der ersten Folge erwähnt haben.

Mit oder ohne Editor

Auch CBASIC, mit dem wir uns in dieser Folge näher auseinandersetzen wollen, wird ohne einen entsprechenden Editor zum Schreiben der Programme ausgeliefert. Wer jedoch diese Serie von Anfang an verfolgt hat, weiß, daß man sich in diesem Fall auch mit dem Schneider Basic-Interpreter behelfen kann. Wichtig ist dabei allerdings noch, daß ein solches BASIC-Programm in jedem Fall als ASCII-File abgespeichert werden muß, damit es auch unter dem CP/M-Betriebssystem gelesen werden kann (s.a. CPC Nr.3). Sollten Sie einen anderen Editor als Wordstar verwenden, vergewissern Sie sich bitte auch unbedingt, daß Ihr Text nicht etwa Steuerzeichen oder Kontroll-Codes, die zunächst noch unsichtbar sind, enthält. In diesem Fall erhalten Sie erst beim Compilieren eine Fehlermeldung, die irgendwo in Ihrem Programm auftritt, und in der Regel keinen Bezug zur eigentlichen Fehlerursache aufweist. Eine weitere Fehlerquelle besteht in diesem Zusammenhang auch beim Arbeiten mit WORDSTAR. Wie bereits in Artikel Nr.2 (CPC Nr.4) erwähnt, muß beim Schreiben von Programmen unbedingt der Modus (N)on Document und nicht (D)ocument gewählt werden. Gefährlich wird es, wenn man mit dem WS-Aufruf auch gleich den entsprechenden Dateinamen übergibt. Jetzt wird automatisch der D-Modus gewählt, was zu den oben bereits geschilderten Folgen führt. Wenn Sie nicht sicher sind, ob Ihre Programm-Datei nicht doch ein paar unsichtbare Steuerzeichen enthält, TYPEn Sie Ihr Programm einfach einmal auf den Bildschirm. Jetzt werden Sie sofort erkennen, ob die Ausgabe nur den reinen Source-Code enthält. Wenn Sie allerdings ein wirres Durcheinander von Sonderzeichen (ab ASCII 128) erkennen, müssen Sie einen Trick anwenden, um Ihr Programm wieder zu reparieren. Geben Sie dazu einfach folgendes ein:

PIP name2.txt = name1.txt [Z]

PIP ist ein sehr leistungsstarkes Dienstprogramm unter CP/M und sollte sich auf jeder Arbeitsdiskette befinden. In unserem Fall bewirkt die Option "Z" beim Kopieren von name1.txt auf name2.txt, daß das höchstwertige Bit (Bit 7), welches ja bei der ASCII-Codierung nicht benötigt wird (< 128), den Wert Null erhält.

Der CBASIC-Compiler

Nun aber endlich zu CBASIC, das sowohl von Markt & Technik als auch von Schneider Data für alle

CPC's angeboten wird, und auch ohne Speichererweiterung lauffähig ist. Da uns die um ca. 25.- DM teuere Version von Schneider Data nicht vorliegt, können wir uns im folgenden nur auf die Version CBASIC 2.0 von Markt & Technik beziehen. Die Aufmachung des Handbuches macht auf Anhieb einen grundsoliden Eindruck und erinnert stark an WORDSTAR, MULTIPLAN oder DBASE aus gleichem Hause. Der in englischer Sprache gehaltene Inhalt ist sehr umfangreich und ausführlich, erfordert jedoch bei Lesern mit schwachen Englischkenntnissen ein wiederholtes Lesen der entsprechenden Kapitel. Daran ändert auch die äußerst knapp gehaltene deutsche Einleitung nicht viel. Wer jedoch schon einmal mit BASIC zu tun hatte, wird sich schnell in dem besonders wichtigen Kapitel 3 zurechtfinden, um sich erst einmal einen Überblick über die vorhandenen Befehle zu verschaffen. Schnell wird deutlich, daß eigentlich alles vorhanden ist, was das Herz eines BASIC-Programmierers höher schlagen läßt und noch einiges mehr. Wer einen CPC 6128 oder Joyce besitzt, braucht nicht einmal auf die Erzeugung von Grafik zu verzichten, da CBASIC die GSX-Software unterstützt, die ja quasi eine Standardschnittstelle für Grafik unter CP/M darstellt. Jeder CP/M-Rechner, der über GSX verfügt, kann daher Ihr in CBASIC geschriebenes und compiliertes Programm ausführen. Es hat jedoch keinen Sinn, GSX z.B. vom 6128 auf den 464 oder 664 zu kopieren, da jede GSX-Software einen Rechner abhängigen Teil enthält, der genau auf das jeweilige System abgestimmt ist.

Funktionen und Unterprogramme

Besonders angenehm fällt auf, daß man bei der Programmeingabe fast vollständig auf Zeilennummern verzichten kann. Allerdings nur fast, da die sehr einfache Struktur von BASIC normalerweise keine echten Unterprogramme mit einer eigenständigen Programmeinheit bietet. So auch bei CBASIC: Der Befehl GOSUB arbeitet ähnlich wie der Befehl GOTO und veranlaßt das Programm lediglich, an einer anderen Stelle fortzufahren. Als "Adresse" muß dann natürlich doch wieder eine Zeilennummer angegeben werden. Eine sinnvollere Verbesserung stellt dagegen die Möglichkeit dar, selbstdefinierten Funktionen, die im Gegensatz zu Schneider-Basic auch mehr als eine Anweisung enthalten können, zu verwenden. Dabei kann man sich sogar den Vorteil der begrenzten Gültigkeit von lokalen Variablen zunutze machen. Was das heißt, wollen wir uns an dem folgenden Beispielprogramm klarmachen, das mit CBASIC compiliert wurde und deshalb keine Zeilennummern besitzt:

```
DEF MULTI(a,b,c) REM FUNCTION
a = a * a
b = b * b
c = b * a
PRINT a,b,c
MULTI = c
FEND                    REM ENDE

REM MAIN
INPUT e,f,g
CALL MULTI(e,f,g) REM 1
PRINT e,f,g
PRINT MULTI(e,f,g) REM 2
END
```

Das Programm besteht aus der Definition einer Funktion mit dem Namen MULTI und dem Hauptprogramm MAIN. Die Funktion MULTI wird von MAIN zweimal aufgerufen. Um den Sinn dieses Programmes zu verstehen, geben wir einfach einmal die drei Zahlen 2,3 und 4 ein. Als Ausdruck erhalten wir dann:

```
 4 9 36
 2 3  4
 4 9 36
36
```

Wir wollen nun versuchen, das Listing und die entsprechende Ausgabe Zeile für Zeile abzuarbeiten. Nach dem Durchlaufen des DEF-Befehls weiß das Programm von der Existenz der Funktion MULTI, die von diesem Augenblick an, ähnlich wie die vordefinierten Funktionen SIN, COS etc., verwendet werden kann. FEND markiert das Ende der DEF-Anweisung. Mit der INPUT-Anweisung erwartet das Programm die Eingabe von drei Zahlen. Durch den nun folgenden Befehl CALL MULTI wird unter CBASIC nicht etwa ein verstecktes Maschinenprogramm gestartet, sondern eine der zwei Möglichkeiten genutzt, um die Funktion MULTI aufzurufen. Dabei werden die sogenannten Formalparameter a,b,c durch die Aktualparameter e=2,f=3 und g=4 ersetzt.
Die in der Funktion enthaltene PRINT-Anweisung veranlaßt den Ausdruck der ersten Zeile. Nach dem Rücksprung in das Hauptprogramm erfolgt der Ausdruck von e,f und g. Sollte man jetzt wieder das Ergebnis von 36 z.B. für die Variable g erwartet haben, die ja in der Parameterliste an der gleichen Stelle wie c steht, wird man erst einmal enttäuscht. Nur beim Einsprung in die Funktion werden die Variablen entsprechend ihrer Reihenfolge in der Parameterliste übergeben, unabhängig von dem jeweiligen Variablennamen, was einem sogenannten "call by reference" entspricht. Beim Rücksprung erfolgt jedoch keine solche Übergabe, so daß die Variablen e, f und g ihre alten Werte behalten. Wir müssen uns daher anders behelfen, wenn wir mit unserem berechneten Wert c auch noch im Hauptprogramm weiterarbeiten wollen. Dazu weisen wir dem Funktionsnamen MULTI die Variable c zu, und nun funktioniert alles wie bei einem richtigen Funktionsaufruf, welcher mit fast der gleichen Syntax auch unter PASCAL und FORTRAN vorgesehen ist. Wieder werden die berechneten Werte a, b, c und in der letzten Zeile endlich der an das Hauptprogramm zurückgegebene Funktionswert ausgedruckt.
Wir haben mit CBASIC also die Möglichkeit, in Funktionen mit sogenannten lokalen Variablen zu arbeiten. Der Vorteil der lokalen gegenüber den globalen Variablen besteht darin, daß sie nur in einem definierten Bereich gültig sind und dadurch keine Variablen mit gleichem Namen im Hauptprogramm ungewollt verändern können. Zur Übergabe eines berechneten Wertes kann man diesen einer Funktion zuordnen. Z.B. MULTI = c. Mehr als ein Wert kann so allerdings nicht übergeben werden. Das geht nur bei echten Unterprogrammen, die z.B. in den Sprachen PASCAL und FORTRAN implementiert sind. Bei PASCAL hat man sogar noch die Möglichkeit, zu entscheiden, ob bestimmte Variablen lokale oder globale Gültigkeit haben sollen, und ob der Wert mit einem "call by reference" oder "call by value" übergeben wird. Was das schon wieder heißt, werden Sie unter anderem in der nächsten Folge erfahren.

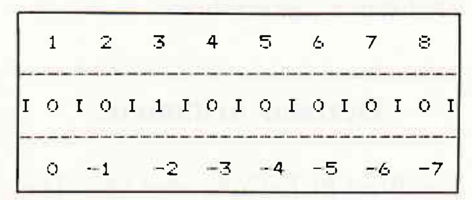
Dezimal-Arithmetik

Das Wort BCD-Code oder auch Dezimal-Arithmetik ist in aller Munde. Nicht zuletzt durch die drei neuen, für die CPC angepaßten Compiler NEVADA-FORTRAN, PASCAL MT+ und CBASIC ist das Thema wieder sehr aktuell geworden. Die 14-stellige Dezimal-Arithmetik bei C-BASIC soll z.B. höchste Genauigkeit bei kommerziellen Berechnungen sicherstellen. Daß es aber auf die 14

Stellen bei CBASIC oder nur acht Stellen bei NEVADA-FORTRAN gar nicht so sehr ankommt, sondern auf die eigentliche Umwandlung einer Dezimalzahl in eine normale Dualzahl oder eben eine BCD (binär codiertes Digit), werden wir Ihnen im folgenden versuchen aufzuzeigen. Dazu geben Sie bitte dieses kurze Listing unter Schneider-BASIC ein:

```
10 a = 0
20 for i = 1 to 100
30 a = a + .33
40 next i
50 print a
```

Wahrscheinlich wird es Sie sehr wundern, wenn Sie als Ausgabe nicht 33 erhalten, sondern 32.9999999. Daß wir unter Schneider-BASIC nur eine Genauigkeit von ca. neun Stellen zur Verfügung haben (s.a. CPC Nr.4), kann eigentlich nicht die Ursache sein, da wir diesen Bereich ja nicht überschreiten. Ändern Sie jetzt bitte einmal die Zeile 30 und ersetzen .33 durch .25. Nun erhalten Sie exakt 25 bzw. 250 oder auch 2500, wenn Sie auch noch die Schleifenzahl jeweils um den Faktor 10 vergrößern. Wenn wir dieses Spielchen aber mit .33 treiben, werden wir eine Ziffernfolge erhalten, die immer mehr von 33 abweicht. Wie kann so etwas möglich sein? Um diese Frage zu beantworten, müssen wir wieder in den Eingeweiden unseres Rechners nachsehen. Wie wir ja bereits in der 2. Folge dieser Serie erfahren haben, legt der CPC seine Real-Variablen in genau fünf Byte ab. Ein Byte ist für den Exponenten vorgesehen und die restlichen vier für die Mantisse. Weiterhin wird die Mantisse "normalisiert", wie bei vielen anderen Rechnern übrigens auch, und liegt daher betragsmäßig immer zwischen 0.5 und 1. Das erste Bit hat den Exponenten 0, das zweite -1, das dritte -2 usw., bis alle 23 Bit verbraucht sind. Sehen wir uns doch einmal an, wie das erste Byte dual codiert aussehen könnte (siehe Bild 1):



1	2	3	4	5	6	7	8
0	0	1	0	0	0	0	0
0	-1	-2	-3	-4	-5	-6	-7

Bild 1: Darstellung von 0.25

Sicherlich wissen Sie schon, welche Dezimalzahl hier dargestellt wird:

$0 * 2^{0} + 0 * 2^{-1} + 1 * 2^{-2} + \ldots$

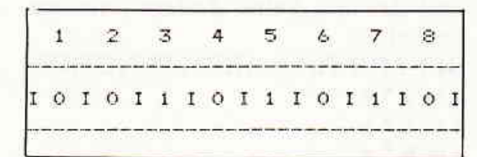
Daraus ergibt sich:

$0 * 1 + 0 * 0.5 + 1 * 0.25 = 0.25$

Die Darstellung der Dualzahl nach Bild 1 entspricht also dezimal 0.25. Nun wollen wir einmal versuchen, die Dezimal-Zahl .33 darzustellen. Dazu müssen wir erst einmal folgende Rechnung aufstellen:

$0 * 1 + 0 * 0.5 + 1 * 0.25 + 0 * 0.125 + 1 * 0.0625 + 0 * 0.03125 + 1 * 0.015625 + 0 * 0.0078125$
$= .328125$

Als Byte würde das folgendermaßen aussehen (siehe Bild 2):



1	2	3	4	5	6	7	8
0	0	1	0	1	0	1	0

Bild 2: Darstellung von 0.33

Wie Sie sehen, reicht ein Byte also nicht aus, um .33 korrekt darzustellen. Das Problem ist jedoch, daß es auch mit den folgenden drei Byte nicht gelingt, diese Dezimalzahl dual genau darzustellen. Immer wird es einen kleinen Rest geben, selbst wenn wir mit double Precision arbeiten würden. Das führt natürlich zunächst einmal nur zu einem kleinen Fehler, der sich jedoch durch das Aussummieren von 100 solcher Fehler, wie in unserem BASIC Beispiel, rasch bemerkbar macht. Das Resultat haben wir ja schon gesehen. Soll eine Zahl also korrekt wiedergegeben werden, muß sie sich als Folge von ganzzahligen Exponenten kleiner oder gleich 0 zur Basis 2 darstellen lassen. Beispiele hierfür sind 0.25, 0.5, 0.328125, etc. Da man seine Variablen natürlich nicht immer so wählen kann, hat man nun das BCD-System ersonnen.

Das BCD-System

Das Prinzip der Zahlendarstellung mit Dezimal-Arithmetik bzw. BCD ist es, jede Dezimalziffer getrennt zu codieren. Jede Ziffer zwischen 0 und 9 benutzt dabei immer soviel Bits, wie zur genauen Darstellung der größten dezimalen Ziffer "9" benötigt werden. Mit 3 Bit kann man nur $2^3 = 8$ Ziffern, also "0" bis "7" codieren. Daher belegt jede Ziffer im dualen Zahlensystem, das auch hier wieder angewendet wird, 4 Bit oder ein Nibble. Da man mit 4 Bit aber $2^4 = 16$ Kombinationen darstellen kann, verschenkt man mit jedem Nibbel sechs Möglichkeiten, was natürlich auf Kosten des Speicherplatzes bzw. der Rechenzeit geht, und nicht immer akzeptabel ist. Zur Darstellung der Dezimalzahl 11 benötigt man z.B. schon 2 * 4 Bit, d.h. ein Byte. Im Schneider Basic wird zur Darstellung der gleichen Zahl nur halb soviel Speicherplatz zur Verfügung gestellt. Bleiben wir jedoch noch bei CBASIC. Mit unserem neu erworbenen Wissen können wir uns leicht ausrechnen, wieviel Speicherplatz eine im BCD-Format codierte Dezimalzahl mit 14 darstellbaren Ziffern benötigt: 7 Byte allein für die Mantisse + 1 Byte für den Exponenten, macht zusammen 8 Byte. Der Exponent wird übrigens nicht in das BCD-Format umgewandelt, sondern normal dual codiert. Da aber nicht nur das Vorzeichen des Exponenten, sondern auch noch das der Mantisse gespeichert werden muß, was jeweils ein Bit kostet, bleiben zur Darstellung noch sechs Bit. Somit ermöglicht diese Darstellung insgesamt einen Wertebereich von $+-(2^6) -1 = 63$, also $+- 10^{63}$. Wenn wir nun wieder auf unser kleines BASIC-Programm zurückkommen und es dieses Mal mit CBASIC kompilieren, ergibt sich auch bei sehr großen Schleifenindizes immer wieder die Ziffernfolge 33.

Anzahl der darstellbaren Ziffern

Wie sich nun die Anzahl der darstellbaren Ziffern (9 bei Schneider BASIC und 14 bei CBASIC) auf ein Ergebnis bemerkbar macht, soll ein letztes Beispiel zeigen:

```
10 a = 1.E9
20 for i = 1 to 1000
30 a = a + 0.25
40 next i
50 print a
```

Unter Schneider-BASIC lautet die Ausgabe für a 1.E9. Beim Addieren der zehnstelligen Zahl und 0.25 wird die kleinere infolge der auf neun Stellen begrenzten Arithmetik einfach abgeschnitten. Unter CBASIC macht sich nun erst der zweite Vorteil dieses Dialektes bemerkbar. Da 0.25 intern problemlos als Dualzahl dargestellt werden kann, wie wir ja bereits oben gesehen haben, könnte man zunächst einmal auf BCD verzichten. Da CBASIC aber bei seinen Arithmetikoperationen 14 Stellen berücksichtigt, lautet das richtige Ergebnis für a 1000000250.

Damit ist auch der 4. Teil dieser Serie beendet. In der 5. Folge soll noch einmal das Thema Unterprogramme aufgegriffen werden, und anhand von Beispielen in BASIC, PASCAL und FORTRAN erläutert werden. Dabei werden wir auch den NEVADA-FORTRAN-Compiler etwas genauer unter die Lupe nehmen.

(Dipl.-Ing. H. Scheruhn)

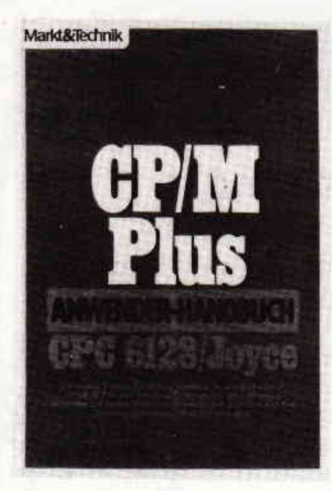
Referenzkarte: Calls & Pokes

Weiter geht es in unserer Referenzkarte mit der Verwaltung des Kernals. In diesem Teil geht es hauptsächlich um Tickerblöcke und synchrone und asynchrone Ereignisse. Wie ein solcher Tickerblock aufgebaut sein muß, und mit welchen Prioritäten einzelne Events ausgestattet sein müssen, wird in den Anhängen unserer Firmwarereferenz beschrieben sein. Sie wollen diese Karten ja später einmal als eine Art Nachschlagewerk benutzen können.

Adresse:	Syntax:	Bemerkung:
&BCF8	Einsprung: HL = Eventblock. Aussprung: AF, BC, DE, HL zerstört.	Lösche ein synchrones Ereignis aus der Warteschlange für synchrone Ereignisse.
&BCFB	Einsprung: keine. Aussprung: Event gefunden: Carry on, HL = Eventblock, A = Priorität. Kein Event: Carry off, A, HL zerstört. Immer: DE zerstört.	Hole den nächsten Event aus der Warteschlange.
&BCFE	Einsprung: HL = Eventblock. Aussprung: AF, BC, DE, HL zerstört.	Bearbeite Event.
&BD01	Einsprung: C = Priorität, HL = Eventblock. Aussprung: AF, BC, DE, HL zerstört.	Stoppe die Bearbeitung eines Events.
&BD04	Einsprung: keine. Aussprung: HL zerstört.	Sperren von synchronen Events.
&BD07	Einsprung: keine. Aussprung: HL zerstört.	Freischalten der synchronen Events.
&BD0A	Einsprung: HL = Eventblock. Aussprung: AF zerstört.	Sperre einen bestimmten Event.
&BD0D	Einsprung: keine. Aussprung: DE, HL = 4 Byte Counter.	Gib die Zeit seit Einschalten des Rechners in 1/300-stel Sekunden zurück.
&BD10	Einsprung: keine. Aussprung: AF zerstört.	Setzen des Time-Counters.

Die Routinen des MC-Packs werden im nächsten Heft beschrieben. Seien Sie sehr vorsichtig im Umgang mit diesen Einsprüngen. Eine genaue Kenntnis der Hardware und der alternativen Registersätze sind zur Handhabung dieser Calls unbedingt vonnöten. Unsachgemäße Handhabung bringt zwar die verblüffendsten Effekte, schicken den Rechner jedoch auch in 99 % aller Fälle auf die Reise ohne Wiederkehr.



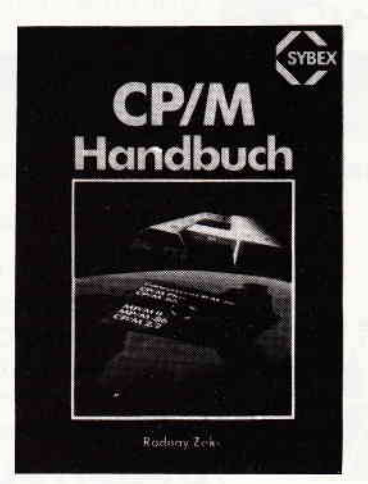


Jürgen Hückstädt
CP/M Plus
Anwender-Handbuch
CPC 6128/Joyce
Markt & Technik 1986
ISBN 3-89090-197-2
Preis: 46,– DM

Endlich ist ein Buch erschienen, das sich (unter anderem) auch explizit mit dem Joyce beschäftigt, genauer gesagt, mit seinem CP/M-Betriebssystem.
Das Buch ist sowohl für Anfänger als auch für Fortgeschrittene geeignet: In neun Kapiteln wird der Leser auf kompetente Weise vom "Adam-und-Eva"-EDV-Grundwissen bis zum "Blick hinter die Kulissen" geführt. Dabei werden die Besonderheiten der beiden im Mittelpunkt stehenden Computer CPC 6128 und Joyce gesondert behandelt; man muß sich also nicht, wie z.B. im dBase-Handbuch, die relevanten Sachverhalte mühsam herausfiltern.
Weiterhin sind die umfangreichen Beispiele, der saubere Druck und die alphabetische Übersicht über die CP/M-Befehle positiv zu erwähnen.
Doch nun etwas Genaueres zum Inhalt:
Die beiden ersten Kapitel sind besonders für CP/M-Neulinge und Aufsteiger von CP/M 2.2 gedacht.
Neben der obligatorischen Einführung in Gerätezusammenstellung und Binärarithmetik werden auch Geschichte und Philosophie von CP/M angesprochen. Danach geht es mit der Praxis und den wichtigsten Verwaltungsutilities weiter. Nach dieser Erklärung der Anwenderoberfläche folgen im 3. Kapitel die "Feinheiten" wie Directoryorganisation, Timestamps und Passwörter. Auch werden User- und Gerätezuordnung, Zeichensätze und Tastaturbelegung erläutert.
Kapitel 4 und 5 sind jeweils den Dienstprogrammen ED und PIP gewidmet, Kapitel 6 beschäftigt sich mit der SUBMIT-Funktion.
Ab Kapitel 7 wird es professionell:
Unter der Überschrift "Debugging" werden die Programme DUMP, SAVE und SID hervorragend erklärt. (Ebenso der Begriff "Debugging in einer hübschen Anekdote.)
Kapitel 8 beschäftigt sich mit der Assemblerprogrammierung, und erläutert die Bedienung der Programme MAC, RMAC, HEXCOM, LINK und LIB. Und so ganz nebenbei hat der Leser die Möglichkeit, sein erstes Assemblerprogramm unter CP/M zu erstellen.
Das 9. Kapitel geht schließlich vollends in die Tiefe.
Neben der leider zu kurz geratenen Darstellung der Speicherverwaltung werden Details wie File- und System Control Block und Systemerweiterungen (RSX) behandelt.
Die Krönung bilden jedoch die detaillierten Darstellungen von BIOS und BDOS-Aufrufen, die zusammen mit Kapitel 8 dem fortgeschrittenen Leser die Erstellung von CP/M-Programmen ermöglichen.
Das 10. Kapitel rundet das Buch mit der bereits erwähnten Kurzbeschreibung der einzelnen Utilities ab.
Obwohl die 42 Einträge sehr kurz gehalten sind, ermöglichen sie dennoch ein zufriedenstellendes Arbeiten mit den Programmen.
Den Abschluß des Buches bilden Tabellen mit Steuer- und ASCII-Codes und ein Stichwortverzeichnis.
Alles in allem ein sehr empfehlenswertes Buch für CP/M Plus-Interessierte, das seinen Preis ganz gewiß wert ist. Zumal es ja angesichts des (Joyce)-Handbuches von Schneider dringend notwendig ist.
Die "Krönung" ist jedoch die Anleitung zu CP/M.
Daß Funktionen wie MAC, SID etc. nicht näher erläutert werden, mag ja im Hinblick auf den potentiellen Anwenderkreis verständlich sein. Daß aber im Anhang auf fast 50 Seiten die Fehlermeldungen überwiegend dieser Programme gelistet werden, ist jedoch – milde gesagt – Unsinn, denn Dank der mangelhaften Anleitung bekommt man diese Meldung jedoch kaum zu Gesicht. Der hier verschwendete Platz wäre besser einer genauen Beschreibung von Dr. Logo zugute gekommen.

Zaks
CP/M Handbuch
Sybex Verlag 1986
7. Auflage
310 Seiten
ISBN 3-88745-053-1
Preis: DM 48,–

Nach wie vor ist gute CP/M-Literatur Mangelware, doch schlägt dieses Buch des weltbekannten Autors Rodnay Zaks viele Brücken. Alles Wissenswerte zu CP/M und MP/M ist hier vereinigt, so daß man schon von einem Standardwerk sprechen kann. Dem Anwender wird dieses Buch nicht nur als Lernmittel, sondern auch als umfassendes Nachschlagewerk dienen.
Anhand vieler Programmbeispiele erläutert Zaks den Umgang und die Programmierung unter CP/M, ohne dabei den Bezug zur Praxis zu verlieren.



Berücksichtigt sind u.a. die Versionen CP/M 2.2, CP/M 86 sowie CP/M plus. Lobenswert ist der didaktisch gut aufbereitete Erläuterungstext, der auch Anfängern keine Probleme bei der Umsetzung auf seinen Rechner bereitet.
Als hilfreich erweisen sich des weiteren die am Schluß des Buches befindlichen Referenzkarten, die übersichtlich angeordnet ein schnelles Nachschlagen gewährleisten.
Ein für alle CP/M-Anwender sehr empfehlenswertes Buch.

Fette
Joyce für Einsteiger
Data Becker 1986
248 Seiten
ISBN 3-89011-170-x
Preis: DM 29,–



Mit diesem Buch präsentiert sich die erste Zusatzliteratur zum Joyce. Viele Anwender haben Schwierigkeiten mit dem Original-Handbuch zu LocoScript, und genau für die kann dieses Buch eine wertvolle Hilfe sein. Allerdings erfährt man hier nichts Neues, sondern vielmehr stellt das vorliegende Buch eine Neuauflage des Handbuches dar. Hier hätte der Autor, statt noch einmal ausführlich über den Anschluß des Joyce-Systems zu berichten, auf fehlende Dinge im Handbuch hinweisen sollen. So wird z.B. mit keinem Wort erwähnt, daß durch Drücken der EXIT- und PTR-Taste eine Hardcopy vom Bildschirm möglich ist.
Überhaupt entsteht der Eindruck, daß hier nicht auf die neue LocoScript-Version 1.21 eingegangen wird, die sich ja in einigen Punkten deutlich

von der alten unterscheidet. Neben LocoScript werden einige Basic- und Logo-Funktionen erläutert, aber auch hier sind erhebliche Lücken festzustellen.
Empfehlen möchten wir dieses Buch nur den Joyce-Besitzern, die große Probleme mit dem Handbuch haben, weil insgesamt die Erläuterung verständlicher geschrieben ist.

Schneider
Wie arbeite ich mit dem Schneider CPC 464
Vieweg-Verlag 1986
258 Seiten
ISBN 3-528-04452-7
Preis: 48,- DM

Die mit diesem Buch angesprochene Zielgruppe sind diejenigen, die nur wenig bzw. noch gar keine Erfahrung mit ihrem CPC 464 haben. Von einer allgemeinen Einführung bis hin zur Erläuterung der einzelnen Befehle mit Beispielen, präsentiert sich das gezeigte Spektrum. Über eine bessere Handbuchversion kommt dieses Buch allerdings nicht hinaus.

Ripota
Alles über Plotter
Signum Medien Verlag 1986
208 Seiten
ISBN 3-924767-08-4
Preis: DM 48,-

Drucker sind nach wie vor die verbreitetsten grafischen Ausgabegeräte. Dennoch gibt es Anwendungsgebiete, für die eine spezielle Grafikausgabe vonnöten ist, weil die Auflösung normaler Drucker hier nicht ausreichend ist. Ein solches Gebiet liegt zum Beispiel im Bereich CAD, wo Konstruktionspläne maßstabsgetreu und hochauflösend erstellt werden. Hier bedient man sich in der Regel eines Plotters. Dieses Buch gibt nun Auskunft über diverse Plotterarten, sowie die Funktionsweise dieser Präzisionsmaschinen. Der Autor versucht, wichtige Entscheidungskriterien für und wider den Kauf eines Plotters aufzuzeigen, und anhand von praktischen Beispielen zu erläutern.
Lobenswert, daß hier nicht nur die rein technische Seite erfaßt wurde, sondern auch auf mögliche Probleme eingegangen wird, wie z.B. das Erstellen eines entsprechenden Treiberprogrammes für die Ansteuerung eines Plotters.
Als hilfreich für einen groben Überblick stellt sich die Marktübersicht der gängigsten Plotter dar, die auch Zubehör und Verbrauchsmaterial berücksichtigt.
Für alle Plotter-Interessierte ist dieses Buch uneingeschränkt zu empfehlen.

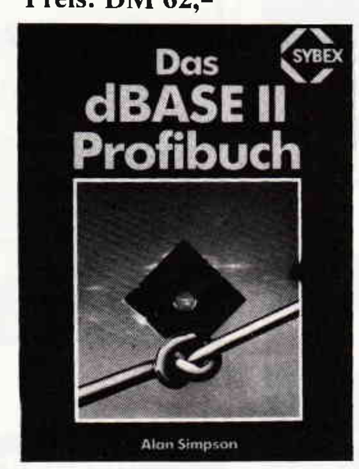
Brückmann/Schieb
Das große Floppy Buch
Data Becker 1986
2. Auflage
422 Seiten
ISBN 3-89011-093-2
Preis: DM 49,-

Das hier besprochene Buch liegt nun bereits in der zweiten überarbeiteten Auflage vor. Für den Umgang mit der DDI-1 erweist sich dieses Werk als unentbehrliches Nachschlagewerk der Diskettenprogrammierung.
Einen besonderen Leckerbissen stellt das umfangreiche und gut dokumentierte DOS-Listing dar, das auch die letzten Geheimnisse des Floppy-Controllers offenbart. Anhand vieler Beispielprogramme erläutern die Autoren den Umgang mit der Floppy und geben sofortige Kontrollmöglichkeiten an die Hand. Daß dabei so wichtige und nützliche Utilities wie z.B. ein Diskettenmonitor, ein kompletter Diskettenmanager



oder eine gute relative Dateiverwaltung in Form von fertigen Programmen veröffentlicht werden, steigert den ohnehin großen Nutzeffekt dieses empfehlenswerten Buches zusätzlich.
Alle im Buch befindlichen Programmlistings können beim Verlag auch auf Diskette bezogen werden, der Preis beläuft sich auf DM 29,-.

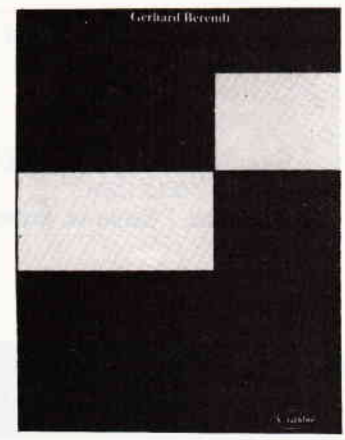
Simpson
Das dBase II Profibuch
Sybex Verlag 1986
428 Seiten
ISBN 3-88745-610-6
Preis: DM 62,-



Mit der Adaption der legendären Programme Wordstar, Multiplan und dBaseII auf die Schneider Rechner, wurde ein weiterer, kommerzieller Einsatzbereich erschlossen. Der Verfasser wendet sich mit diesem Werk an die Anwender, die ihre betriebliche Datenbank-Verwaltung noch effektiver gestalten wollen. Daraus ergeben sich auch die behandelten Themengebiete, die sich vorwiegend mit dem dBase-Einsatz im kaufmännischen Bereich beschäftigen, wie etwa Inventarlisten, Buchhaltung etc. Das beginnt bei der sinnvollen Erstellung der Eingabemasken und setzt sich in Form von Programmen für geschäftliche Anwendungen fort. Sehr gut gelungen sind die jeweiligen Erläuterungen, die einen schnellen und sicheren Einstieg in Programmiertechniken vermitteln. Die dabei erarbeiteten Routinen können in eigene Programme eingebaut werden; es präsentiert sich eine ansehnliche Sammlung von grundlegenden Routinen. Für alle dBaseII-Anwender öffnet sich eine reichhaltige Bibliothek, die den Umgang mit dem Datenbanksystem erleichtert. Kenntnisse der dBase-Programmierung sollten allerdings vorhanden sein.

Berendt
Turbo Pascal auf CP/M
Elektra Verlag 1986
143 Seiten
ISBN 3-922238-63-7
Preis: DM 34,-

Wie schon öfter erwähnt, ist Pascal (insbesondere Turbo-Pascal) dabei, zur Standardsprache auf 8- und 16-Bit-Rechnern zu werden. Das hier vorliegende Buch wendet sich nun an diejenigen, die bereits mit den Grundzügen von Turbo-Pascal vertraut sind.
Der Autor zeigt anhand von einfachen Beispielen, wie Prozeduren und Funktionen unter Turbo-Pascal Verwendung finden. Großen Wert legt der Verfasser auf die verständliche Erläuterung der Zugriffsmöglichkeiten auf interne CP/M-Routinen, die von Turbo-Pascal heraus erreichbar sind. Allerdings ist nur das Betriebssystem CP/M 2.2. berücksichtigt.



Der Leser bekommt mit diesem Buch ein Werk in die Hand, das ihm ein effektives Programmieren erleichtert, ohne daß dabei Assembler-Routinen entwickelt und eingebunden werden müssen. Neben Kenntnissen in Turbo-Pascal sollten allerdings die Grundzüge des Betriebssystems CP/M hinreichend bekannt sein.

Backnang



Bad Reichenhall



Ihr Spezialist für DFÜ (RS 232)
Peter Kohl
Waaggasse 4, 8230 Bad Reichenhall
Tel.: 08651/66773

Berlin

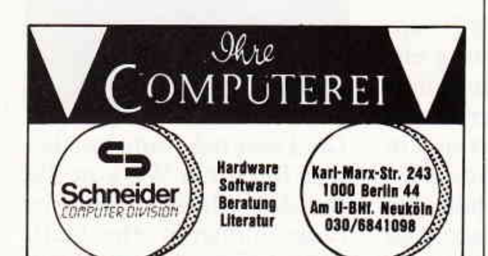
Ihr starker Schneider-Partner



Schöneberger Straße 5 • 1000 Berlin 42 • Tel. 030-782 91 80/60



A+C Vertrieb, 1 Berlin 44, Emser Str. 18



Elektr. + elektronische Geräte, Bauelemente + Werkzeuge
ELECTRONIC VON A-Z
Stresemannstr. 95 · Berlin 61
Telefon (030) 2 61 11 64



Bonn

RADIO-FERNSEHEN HIFI-VIDEO **Schäfer**
SERVICE SERVICE SERVICE SERVICE
Plittersdorfer Straße 206 Telefon 36 40 29

Bremen



Für Einsteiger, Aufsteiger und Umsteiger!

Home- und Personal-Computer von Adler, Atari, BASF, Casio, Commodore, Epson, IBM, Sanyo, Schneider, Sharp, Spectra-Video, Taxan, Texas Instruments und Zenith.

Wir bieten Hardware, Software und sorgfältige Beratung.

Obernstr. 82-88 · 2800 Bremen 1
Telefon (04 21) 3 09 90

Castrop-Rauxel

EINE GUTE IDEE NACH DER ANDEREN
Schuster Electronic
COMPETENT IN SACHEN COMPUTER & ELECTRONIC
ELEKTRONISCHE BAUELEMENTE ALLER ART



Obere Münsterstr. 33 4620 Castrop-Rauxel (02305) 3770

Dortmund



Für Einsteiger, Aufsteiger und Umsteiger!

Home- und Personal-Computer von Adler, Atari, BASF, Casio, Commodore, Epson, IBM, Sanyo, Schneider, Sharp, Spectra-Video, Taxan, Texas Instruments und Zenith.

Wir bieten Hardware, Software und sorgfältige Beratung.

Nur Westenhellweg 102-106
4600 Dortmund · Tel. (0231) 184 51

Düsseldorf

...alles für die Schneider-Computer!

Computer Terminal





Freiburg

computer aktuell

Südbadens kompetenter Computer-Partner.

Kaiser-Joseph-Str. 232
7800 Freiburg, Tel.: 0761/2180225

Hamburg

Jetzt auch bei uns:
Joyce und CPC 464 + 6128

Schneider COMPUTER DIVISION
autorisierter Fachhändler

Anwenderprogramme z.B. für Joyce: Wordstar 3.0, d Base II, Multiplan, Finanzbuchhaltung, Business Pack, DR DRAW, DR GRAPH, Schach

Große Auswahl an Spiel- und Anwenderprogrammen, Zubehör und Literatur für SCHNEIDER und COMMODORE.

Gärtnerstr. 5 · 2000 Hamburg 20 · Tel. 420 46 21



Hannover

Löhne/Ostwestfalen

Nürnberg

Rüsselsheim

Spesenroth

Würzburg

Schweiz

Zürich

Österreich

Traun

In dieses Händlerverzeichnis können sich alle
Computer-Händler
eintragen lassen
Für telefonische Anfragen steht Ihnen Herr Schnell gern zur Verfügung.
Tel.: 05651/8702.

Achtung! An alle CPC-User-Clubs!
Suchen Sie noch Mitglieder oder wollen Kontakte zu anderen CPC-Usern oder Clubs knüpfen? Dann schreiben Sie uns doch einfach. Wir veröffentlichen jede Anschrift eines CPC-User-Cubs gratis! Falls auch Ihr User-Club sich einmal vorstellen möchte (evt. mit Foto), nutzen Sie unsere Zeitschrift als Verbindungsglied und Kontaktadresse!

Amstrad-Club Schweiz
Jetzt auch für CPC 664 und 6128!
Kurt In-Albon,
Mühlackern,
CH-3931 Eggerberg

Clubgründung CPC 6128
im Raum Hanau-Gelnhausen
Info unverbindlich bei:
Richard Luppert,
Gartenstraße 18,
6456 Langenselbold,
Telefon: (0 61 84)23 07

JOYCE-Profi-User-Club
– Gründung geplant –
Mitglieder in ganz Deutschland und im skandinavischen Raum gesucht.
Infos und Kontaktadresse:
Günter Zachritz,
Scharnhorststraße 2,
2300 Kiel 1

CPC-User-Club
im Raum 6250 Limburg?
Wer Lust hat, bitte melden bei:
Telefon: (0 64 36) 34 85

Clubgründung im Raum Bergheim/Erft angestrebt
Interessenten bitte melden bei:
Rudolf Nitsche,
Telefon: (0 22 71) 6 55 88,
täglich ab 16.30 Uhr

Berliner Joyce-Benutzer
ruft doch mal an!
Richard Suchenwirth,
Telefon: 3 91 47 80

CPC Kontakt
Tausch von Originalsoft- und Hardware. Kein Beitrag!
Ralph Lukas,
Im Junkerbruch 3,
5143 Wassenberg 4,
Telefon: (0 24 32) 34 98

UCC-GREFRATH
sucht Kontakt zu anderen Clubs
Bitte melden bei:
Carsten Hoener,
Velourstraße 19,
4155 Grefrath 1

CPC-Club Esslingen
sucht Mitglieder in ganz Deutschland; kein Mitgliedsbeitrag!
Jörg Bauer,
Kirchstraße 32,
7441 Unterensingen

Schneider CPC Club Bayreuth und Bayreuther Computerclub
Wir suchen Mitglieder aus ganz Europa
Bitte melden bei:
Marco Konrad
Fontanestraße 4
8580 Bayreuth

CPC Clubgründung angestrebt
für den Raum Rüsselsheim
Bitte melden bei:
Markus Jöst
Amorbacher Straße 52
6090 Rüsselsheim/Königstädten

Suche Anschluß an Schneider CPC-Club
im Raum Unna oder Kamen
Friedhelm Schürmann
Schäferstraße 21 a
4708 Kamen
Telefon: (02307) 74653

Club zum Null-Tarif
Heinrich Lauterbach
Telefon: 0 21 04 / 4 26 91

JOYCE-Userclub
Clubgründung geplant
Heinz Haupt, 4300 Essen 1
Fridtjof-Nansen-Straße 1
Telefon: (0201) 411020

Schneider-Computerclub e.V.
Abt. Ostwestfalen/Lippe
Info: Computer-Freunde/Löhne e.V.
Postfach 1101
Telefon: 05732/3396
4972 Löhne 1

IAC – International Amstrad Club
Interessengemeinschaft für Computeranwendung
Kölner Straße 66
5630 Remscheid 11
Telefon: 02191/68571

Kontakt zu CPC 6128-Besitzern im Raum Traunstein gesucht
Leopold Kramert
Herbsdorfer Straße 18
8221 Aiging
Telefon: 0 86 69 / 61 35

Styria CPC 464 Freaks
Info anfordern bei:
Michael Rothauer
Johann-Strauß-Weg I
8641 St. Marein im Mürztal
Steiermark Österreich

FREAK-SOFT-CLUB
CPC 464:
Ralf Hopperdietzel
Theodor-Storm-Straße 22
8670 Hof (Saale)
CPC 664, CPC 6128:
Michael Eckert
Zobelsreuther Straße 30
8670 Hof (Saale)

CPC-464-Userclub Tübingen/Böblingen
sucht noch interessierte CPC-User
Kein Clubbeitrag!
Kontakt:
Andreas Kopp
Harbigstraße 79
7030 Böblingen
Telefon: 07031/26508

Amstrad-Schneider-Userclub
Aachen **A.S.U.C.A.**
nimmt noch Mitglieder auf
Dipl.-Ing. R. Cloots
Zeißstraße 7
5132 Übach-Palenberg
Telefon: (02451) 46608

CPC-Clubgründung angestrebt im Raum Erftkreis (nähe Bergheim)
Ab 16 Jahre aufwärts
Bitte melden bei:
Rudolf Nitsche
Telefon: (02271) 65588

Berliner User-Club
sucht Mitglieder, auch weibliche, jeden Alters
Infos vom:
Berliner User-Club
Wolfgang Windorpski,
Gritzner Straße 38,
1000 Berlin 38,
Telefon: (030) 8 22 77 50 ☎
(Anrufbeantworter)

Biete an Software

Verk. Original Software

Rechnungsformular	Kass. 15,– DM
Fighter Pilot	Kass. 15,– DM
Lagerverwaltung	Kass. 30,– DM
CPC Text/Adress.	Kass. 30,– DM
Winter Sports	Disk. 30,– DM

Telefon: 06331/76179

NEUESTE JOYCE-PROFI-SOFTWARE:

Joyce/J-Plus	DM 1799/2480,–
Prompt Dateigenerator	DM 69,–
Adressen-, Kasse	je DM 98,–
Lagerverwaltung	DM 198,–
Finanzbuchhaltung	DM 194,–
COBOL-Comp. dt. Handb.	DM 129,–

MicroM.-Worms, 4780 LIPPSTADT
JOH.-WESTERMANN-PLATZ 1

★ ★ **SCHNEIDER-CPC-SOFTWARE** ★ ★
★ Die neueste Anwendersoftware ★
★ aus England für alle CPC's ★
★ auch für JOYCE ★
★ Info (1,– DM) anfordern bei: ★
★ UNICOM COMPUTERTECHNIK ★
★ Postf. 210405, 4100 Duisburg 1 ★

★ ★ ★ **SCHNEIDER CPC 464** ★ ★ ★
★ **Hannover's** ★
★ **SOFTWARETHEK Nr. 1** ★
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

Mastertronic-Games	a.C 9.90
Warlord-Centre Court	a.C 29.—
Music Composer/Death Pit	a.C 32,—
Fighting Warrior	a.C 36,—
Hyper Sports	C/D 42/54,—
The way of the Tiger	C/D 36/52,—
Yie ar Kung Fu	D 52,—
Colossus 4 Chess	D 59,—
Hi Rise/The Devils Crown	a.D 52,—
Textomat plus CPC 6128	D 198,—
GRAPHICS Lightpen+Software	98,—

sofort **CSJ NEWS** anfordern
★ ★ **CSJ COMPUTERSOFT JONIGK** ★ ★
★ **An der Tiefenriede 27, 3000 Hannover 1** ★
★ ★ ★ **Tel.: 0511/886383** ★ ★ ★

JOYCE 8256
Software für alle prof. Anwendungen z.B. DATENBANK, FIBU, AUFTRAG, PLAN, MERGE **max.** à DM 100,–
G.J. Rau, Regensburgerstr. 33, 8500 Nürnberg-30, Tel.0911/466866

Komplette Hausverwaltung
Es können 1 – 150 Wohne. verwaltet werden. Prog. enth. auch komp. Heizkostennebena., Mahnschr., Abbuchung aller Kosten mit Dat. und vieles mehr. Info: K. Frank, Friedhofstr. 26, 7101 Löwenstein

Für Schule und Lehrer!
Notenverwaltung: Berechnung mit Wichtung. Notenlisten - Schnitt - Übersichten - Klassenlisten - Notenspiegel - Unk.-Beitr. 65,– DM für Disc. 3". Info Freiumschlag, Claus Bernhold, Rommelstr. 31, 8783 Hammelburg 2

HOAI-Honorare: Spitzenprogramm
H. Bernards, Tel.: 02223/23292

Verkaufe: Bruce Lee, Hacker, u.v.a., nur Disc., Tel.: 09375/357

KONTO 2000-Geldv.	C 49,–/D 59,–
STAR-TEX Textverarb.	Disc. 79,–
TEXT-SUPER-5 Textv.	C 29,–/D 39,–
ECONOMIC-CAR	C 29,–/D 39,–
BÖRSENSPIEGEL	C 29,–/D 39,–

Informationen: Th. Wieland
(D=Disc.) Falkertstr. 71
(C=Cass.) D-7000 Stuttgart-1

Grünmonitor GT 64 für 464 190,– ★ NLQ 401 m. Traktor 580,– ★ Standard Basic Buch 30,– ★ Multi Datei 55,– ★ Basic-Kurs I K 45,– ★ Cyrus II Chess K 28,– ★ Fighter Pilot K 28,– ★ 50 Kaskade-Spiele K 40,– ★ Alles Originale ★
Telefon: 02150/2226

Adressenprogramm, bis 150 Adr. Für CPC 464 mit Grünmonitor 30,– DM, auf Cass., L. Schittko, Silker Weiche 30, 2057 Reinbek

CPC-Int. 85+Pr.-Kass., T.: 07935/8470

SOFTWARE-KNÜLLER
Für CPC 464/664/6128/Joyce
Kostenlose Liste anfordern!
Jürgen Merz-Elektronik,
Lengericher Straße 21,
4543 Lienen,
Telefon: 05483/1219 o. 8326

TEXPACK NEU 664-6128, DM 120,–
Zmijewski, Postfach 1211, 8031 Gilching, Tel.: 08105/4873

Gute Programme für den 464.
Hauptsächlich Anwendungen! Info gegen Rückporto bei P. Grigorov, Hans-Endt-Str. 126, 4 Düsseldorf 13

CPC-Schneider Flugtraining für CPC 464, 664, 6128. Umfassende Bewertung der Flüge.
A) Hubschrauber-Simulator in Aktion. 9 Anzeigen im Cockpit. 3 Flugprogr. zur Wahl 29,– DM
B) Space Shuttle-Landung Echtzeitsimulation, nach NASA-Unterlagen. 29,– DM
C) Boeing-727 Simulator. Dies Programm ist zur Anfänger- und Instrumenten-Flugschulung geeignet. Mit Anleitung 34,– DM
Ab 2 Program. jedes minus 5,– DM. Auf Disk plus 7,– DM. Info gegen Rückporto. Fluging. F. Jahnke, 3344 Flöthe 1, Tel.: 05341/91618

Org. TURBO PASCAL m. 400 S. Handb.
VB 130; J. Rüchardt, T.: 0228/637225

Biete Chemieprogramm Cass. 10,– DM Disc 25,– DM, tausche auch Software Geld. o. Liste an: H. Groß, Ginsterberg 13, 6630 Saarlouis 4

Verkaufe o. tausche 100 Top-Spiele
z.B. Elite/Sorcery/Hobbit/Way of the Tiger/Exploding Fist/NeverEnding St. usw. , Preis nach Vereinbarung; Knaus M., Tel.: 09503/1641

● **ACHTUNG – BAUINGENIEURE** ●
Iterative Wasserspiegellagen-berechnung für offene Gerinne.
Cass. 175,– DM, Disk. 180,– DM inkl. MwSt. u. Versandkosten
Info: Tel.: 0651/62968 o. 61990 oder gegen 1,– DM in Briefmarken

Wordstar (3" f. CPC) m. Handbuch, orig.-verpackt VB 170, Tel.: 02203/82988

WORDSTAR f. CPC 6128, von M+K, für DM 110,– zu verkaufen.
Ruf: 02241/53246

Wegen Systemaufgabe verk. Orig.
Anwender Programme auf Disk.
Quick-Calc Buchhaltung und Quick-Word Textverarbeitung beide Prog. auf 1 Disk.
NP 376,– DM für 199,– DM und 1 Disk. Fibuking Finanzbuchhaltung NP 136,– DM für 80,– DM.
Tel.: 06331/76179.

Programme aus Eigenentwicklung
für alle CPC's preiswert von Friedrich Neuper, Postfach 72, 8473 Pfreimd. Gratisinfo anfordern.

DIAS ordnen mit dem CPC
bis 30000 Dias; Suchzeit ca. 1 Sekunde. Info geg. Rückporto W. Grotkasten, Birnenweg 6, 7060 Schorndorf, Tel.: 07181/42846

BRD – Wie gut kennen Sie Deutschland?
Testen Sie sich bei diesem interessanten und lehrreichen Spiel. Ideal für Schulen. Disc. 3" DM 40,– (incl. Versandk.)
Schein oder Scheck.
R. Misch, Kirchdorferstraße 20, 7951 Oberopfingen

■ **JOYCE** ■ **LocoScript-Fibel** ■
Präzise, systematische, einfache Anleitung. 19,– DM + Porto + Nachnahme. Walter Ehlers, 2000 Hamburg 56, Rheingoldweg 5, Tel.: 040/813721

FCM-85 für CPC 464.
Alle Informationen über Dateien u. Programme von Floppys u. Cassetten. Automatische Erfassung, Speicherung, Kommentierung und vieles mehr.
FCM-85 mit ausf. Handbuch D 64.90
Kabellänge nach Wunsch
Druckerkabel (CPC 464, 1.50 m) 48.90
je weit. angef. 20 cm (max. 3 m) 1.40
Stereokabel (Cinch o. DIN, 1.50 m) 14.90
je weit. angef. 50 cm (max. 10 m) 1.40
Versand gegen V-Scheck oder per NN. (zzgl. 5.– DM Versandpauschale)
Farbband-Regenerierung für NLQ 401
Der Hit. Schicken Sie uns Ihr altes Farbband. Für nur 10,– DM (Schein oder V-Scheck) erhalten Sie Ihr Farbband in Topqualität regeneriert (schwarz oder blau) zurück.
Infos gegen Freiumschlag in 1er Wochel
Siegmund Braun, Tel.: 08151/2331 o. 2524, Andechser-Str. 54, 8130 Starnberg 2

Die Sonne geht auf ...
Supersoftware – Superpreise:
Manic Miner 9.90
Ping Pong 27.90
Rambo 25.90
Hacker 29.90
Elite (dt. Version) 52.90
CYRUS 3-D Chess 34.90
und noch vieles mehr!!
3-D Clock Chess f. JOYCE 62.90!
Box f. 40 3"-Disk. nur 33.90
Komplette Liste von
SUNSHINE-Software · A.d. Schilde 14
5270 Gummersbach · T. 02261/75752
★ Softwareautoren gesucht!! ★

Biete an Hardware

★ ★ **SCHNEIDER-SOMMERHITS** ★ ★
★ **DFÜ-SET** komplett **DM 348,–** ★
★ best. aus Akustik-Koppler, ★
★ Interface und Terminalpgm. ★
★ **VORTEX-Speichererweiterung** ★
★ SP-128 nur **DM 299,–** ★
★ SP-256 nur **DM 399,–** ★
★ **VORTEX-Floppystation** ★
★ **F1-X** 5,25"/708kB **DM 688,–** ★
★ **3"-Floppylaufwerk** **DM 199,–** ★
★ INFO's gegen 2,– DM Porto ★
★ **SOFTWARE-SERVICE-RETHEMEIER** ★
★ Vlothoer Str. 65a, 49 Herford ★
★ Telefon: 05221/86798 ★

CPC 464 Grün + Drucker (CPA-80) + DDI + 3 DB-Bücher + div. Software, NP ca. 3200,– DM, 1a-Zustand für VB 1800,– DM zu verkaufen. Tel.: 04852/51369 ab 18 Uhr

Lichtgriffel mit Programm für CPC 464/664 nur **DM 49,–** Versand gegen Scheck/Nachnahme Info gratis. Fa. Schißlbauer, Postfach 1171S, 8458 Sulzbach, Tel.: 09661/6592 bis 21 Uhr

Cumana 3"-Floppy B f. CPC 6128
3 Mon. alt, VHB 350,–; 02203/82988

CPC 464 m. Farbmonitor Druckerschnittstelle m. 8-Bit; VB 950,–, Tel.: 05383/1725

ACHTUNG SCHNEIDER 464 incl.
Bücher und ECBPL. DM 650,–
Vortex-Speichererw. 2x64K DM 250,–
Vortex Diskst. DM 1500,–
RS 232 DM 130,–
IBM-Gehäuse DM 400,–
Cobol-COMP. DM 100,–
Div. Zubeh. alles VHP Tel. 0681/45539

Verk. SCHNEIDER NLQ 401
Umstänaehalber, in TOP-Zustand; mit Form.-Traktor, Ers. Farbband VHB 600,– ☎ 06101/3609 ab 17 Uhr

CUMANA 3" Zweitlaufwerk für CPC 6128 (2 Mon. alt) DM 320,– orig. DEISYS (rel. Dateiverwalt.) NP 198,– für DM 120,–; Günter Treusch, Schnaingartenstr. 2, 6070 LANGEN, Tel.: 06103/22287

Floppy DDI-1 VB 580,– DM, CPC 1 Jahrg. komp. 70,– DM, Tel.: 05371/61680

Verkaufe CPC 464 + Literatur + Software! Preis VHS; Tel.: 07263/5290

Verk. wegen Systemwechsel CPC 464 m. Monitor (grün) für DM 450,– VB Telefon: 06331/76179

36CM Farb-TV-Monitor f. CPC ★ RGB + AV-Eing. ★ 80 Z.-Darstell. ★ Hochauflös. Bildröhre ★ 16 Prog.+Femb. DM 699. m. eingeb. Netzteil 5V+12V DM 799. Huber & Co., 7601 Ortenberg, ☎ 0781/34162

CPC 464 Grün 550,–, R. Becklein, 7189 Unterspeltach, HSNr. 38

CPC 464 Color + dk'tronics Lightpen + 7 Bücher v. Data Becker (Neupr. 300,–) + Spiele org. (Neupr. 350,–) Joy+ Sonderhefte + Soft auf weitere 20 Cass. + 100te v. Listings aller Art zus. in einem Ordner + Hefte Schneider Int. ab 3 – 12/85 kompl. für 1700,– Wenn gewünscht, kann auch Grafpad 2 ART mit übernommen werden, Preis 200,–. Tel.: 09342/84112

★ ★ ★ **ECB-Bus für alle CPC's** ★ ★ ★
Machen Sie aus Ihrem CPC einen **professionellen** ECB-Rechner für universel. Einsatz mit unserer ECB-CPC Interfacekarte z. Steuern, Regeln, Datenerf., Roboter steuern etc. Fa. **SIKOS**, Neuwerkerweg 17, 8504 Stein, Tel.: 0911/325558

CPC 464 – Verkauf einer Komplettanlage wegen Systemwechsels.
CPC 464 mit Farbmonitor, DDI-1, Star Gemini-10X, Staubhauben, 4 Data Becker-Bücher + Fachlit. Diskettenprogramme original Texpack + Topcalc. VB DM 2700,– NP 3800,–, Tel.: 05130/7360

CPC 464-System komplett mit Farbmonitor, massenweise Software und Literatur, alles in ausgezeichnetem Zustand. Neupreis komplett Sfr. 1750,–, Verkaufspreis Sfr. 1150,–; Walter Wiederkehr, Unterdorf, CH-6243 Egolzwil, Tel.: 045/713215, Di. – Fr. ab 19.30 Uhr

Verk. CPC 464 + Grünmonitor + Bücher + Software + Zeitschr. Preis 750,– DM, Tel.: 0421/593227

VORTEX F1-S und F1-D
5.25", 708kB bzw. 1. 4MB Floppy wie neu! Tel.: 07141/64637

CPC 464 Grün + DD1 neuwertig
DM 1000,–, Wochenende 07977/653

Höchstpreise für Ihr Altgerät
bei Computerneukauf. Comp.-Vers. Trier, Postf. 2106, Tel.: 0651/16366

Steuern – Messen – Regeln
mit E.B.S. Schnittstellenmodulen Kostenloses Info anfordern. Griesmayr electronic Josef Griesmayr Musenbergstraße 33 8000 München 81 ☎ (089) 95 34 03

Seikosha 700 Farbdrucker ~ 500,–
Phase 4 CP/M-Text+Mailm. ~ 120,–
12 Bücher, Data-B., Heim, (Basic, Floppy, CP/M, M-Code...) ~ 150,–
Lightpen+Softw.+Joystick ~ 75,–
CPC 464+Spiele+Abdeck. ~ 450,–
Anacker, T. 06131/226759, ab 15 Uhr

Suche Software

Es programmiert für Sie auf Schn.-CPC's
N. Neuhaeuser, Steinstr. 76, 4223 Voerde 1, ☎ 02855/5807 ab 19.00 Uhr

Schulsoftware gesucht für Schneider CPC 6128: Stundenplanerstellung, Bundesjugendspiele, Schulverwaltungsprogramme etc. Angebote an A. Diesing, Tel.: 07581/2493

Suche Hausverwaltungs-Software
für Schneider Joyce ☎ 04925/1499 ab 18 Uhr

Suche Hardware

DDI-1 kplt. Alex Jansen
Dr.-Fritz-Huber-Straße 74
8090 Wasserburg, Tel.: 08071/8910

Geschäfts-verbindungen

Erstellung von Handbüchern und Übersetzungen; Call: 0221/384849

Stellenangebote

Computer-Lehrer mit Draht zu K. u. Jugendl.
Raum Norddeutschland sowie Raum Südbaden gesucht auch Hardw.-Spezis u. Freiz.-päd. Tel.: 04871/1274, 040/861255

Das ist Ihre Chance... schon eine Kleinanzeige bringt oftmals großen Erfolg und hilft neue Kontakte knüpfen. Nutzen Sie unser Angebot und profitieren Sie von der Tatsache, daß unsere Zeitschrift „Schneider CPC International" jeden Monat von mehreren zigtausend Computer-Interessierten gelesen wird.

Möchten Sie etwas verkaufen, tauschen oder suchen Sie das „Tüpfelchen auf dem i" – dann sollten Sie die eigens hierfür bestimmte Bestellkarte im Heft ausfüllen und an unseren Verlag absenden.

Ihre Annonce erscheint dann in der nächsterreichbaren Ausgabe.

Wir möchten ausdrücklich darauf hinweisen, daß wir keine Anzeigen veröffentlichen, aus denen ersichtlich ist, daß es sich hierbei um Veräußerungen von Raubkopien oder dergleichen handelt!

Des weiteren machen wir darauf aufmerksam, daß indizierte Computerspiele nicht in Form von Anzeigen beworben werden dürfen.
Die Redaktion



Souls of Darkon

Vor langer Zeit gab es irgendwo in der Weite der Galaxis einen Planeten, auf dem man die Künste der Magie ebensogut beherrschte, wie die weit entwickelte Wissenschaft. So kam es zu der für uns nur schwer vorstellbaren Situation, daß Zauberei und Naturgesetze für die Menschen, die dort lebten, zwei gleichbedeutende Dinge waren.

Aber wie auch in der Wissenschaft gibt es auch in der Magie Bereiche, in die man besser nicht vordringen sollte.

Immer wieder fanden sich Menschen, die bereit waren, für Macht und Wissen sogar ihre Seele zu verkaufen. Doch konnte das Übel, das dadurch heraufbeschworen wurde, immer wieder unter Kontrolle gebracht werden. Man versuchte, die Studien der Schwarzen Magie unter Strafe zu stellen. Aber selbst drakonische Strafen führten zu nichts.

So hoffte man, daß es den Meistern der klaren Künste auch in Zukunft gelingen möge, all das böswillg heraufbeschworene Übel zu bannen.

Irgendwann einmal geschah es dann aber, daß ein verderbter Geist eines üblen, bösen alten Buches habhaft wurde. Einem Buch aus der frühesten Vergangenheit des Planeten, geschrieben von Wesen, die Äonen vor dem Menschen diesen Planeten beherrschten. Längst ausgestorben, doch ihre satanische Ausstrahlung war durch das Buch hindurch bis in diese Tage erhalten geblieben. Als der verblendete Adept dann diese verfluchten Beschwörungen flüsterte, befreite er etwas, das seit Abermillionen Jahren in der Verbannung geruht hatte. Darkon, einer der dunklen alten Götter, war zurückgekehrt, und in den endlosen Jahren seiner Gefangenschaft hatte er nur einen Gedanken in seinem unfaßbar bösen Hirn: Rache. Rache an denen, die ihm diese Gefangenschaft auferlegt hatten.

Rache an den Menschen, deren frühesten Vorfahren es gelungen war, Darkons Macht zu brechen. Doch aus diesen unendlich lange zurückliegenden Tagen gab es nichts; keine Aufzeichnungen, aus denen hervorgegangen wäre, wie Darkon zu besiegen sei. Und so fand sich auch niemand, der Darkon die Stirne bieten konnte und dies überlebt hätte.

Als erstes bemächtigte sich Darkon der gewaltigen Robotarmeen, deren eigentliche Bestimmung das Beschützen von Menschen war. Mit ihrer Hilfe begann beispielloser Vernichtungsfeldzug gegen die Menschheit.

In diesen Tagen hielt der grimme Schnitter reiche Ernte unter den Menschen von Darkon. Aber während Darkon nur eine magische Macht besaß, hatten die Menschen noch die Wissenschaft. Mit ihrer Hilfe gelang es schließlich, die Roboterheere stillzulegen. Trotz all seiner Macht gelang es Darkon nicht, das so herbeigeführte Patt zu beeenden. Doch immer noch lebten die Menschen allerorts in Furcht vor seiner abgrundtiefen Bosheit. Aber eines Tages, so weiß eine alte Prophezeiung zu berichten, wird einer kommen und Darkons Macht brechen, ihn in den dunklen Pfuhl zurückschleudern, aus dem er gerufen worden war.

Wer weiß, vielleicht sind Sie derjenige, auf den die Menschen dort warten. Jener, dem es gelingt, das Wissen zu erlangen, mit dem Darkons Einfluß ein für allemal gebrochen würde – wer weiß?

Meines Wissens ist SOULS OF DARKON das erste Abenteuer des englischen Software Hauses Taskset, die bisher wohl besser als Produzenten von Arcade-Spielen à la Pipeline bekannt waren. Um es vorneweg zu sagen: dafür, daß es sich hier sozusagen um ein Debut handelt, ist es ein erstaunlich gutes Abenteuerspiel geworden. Auch in diesem Falle war man der Meinung, daß es wohl besser ein grafikunterstütztes Spiel sein sollte, hat aber nicht den Fehler begangen, der Grafik zu viel Platz einzuräumen. In diesem Zusammenhang ist auch einiges an den jeweiligen aktuellen Bildschirmdisplays interessant. Prinzipiell ist der Screen vierteilig. In einem Fenster, das jeweils ein Viertel des

Monitors einnimmt, wird ein Bild der Örtlichkeit angezeigt. Zwar nur zweifarbig und allem Anschein nach aus Zeichensätzen definiert, aber recht gut gelungen und sehr effektiv. Der zweite Bereich befindet sich quer über den obersten sechs Zeilen, und gibt eine permanente Beschreibung des Ortes ab. Die mittleren fünfzehn Zeilen sind zweigeteilt: In der linken Hälfte befindet sich das Bild und in der rechten der Kommunikationsteil, in dem der Computer seine jeweiligen Antworten präsentiert. Im vierten Bereich, der die untersten zwei Zeilen des Monitors ausmacht, befindet sich dann das eigentliche Eingabefenster, in das man seine Aktionen eingibt. Bedauerlicherweise setzt auch SOULS OF DARKON den Trend der Abenteuer ohne eine echte Help-Funktion fort. Versucht man trotzdem, einmal Hilfe vom Computer zu bekommen und tippt "Help" ein, antwortet dieser, daß man es doch einmal mit Vocab oder mit Legend versuchen sollte. Bei Vocab bekam ich eine recht ansehnliche Liste von Wörtern gezeigt, die SOULS OF DARKON versteht und mit Legend veranlaßt man den Rechner, eine Geschichte zu erzählen, die mit dem Ort, an dem man sich gerade befindet, in dierektem Zusammenhang steht. Zwar stellen diese Geschichten in einer gewissen Art und Weise auch eine Help-Funktion dar, können diese allerdins nicht ersetzen – dazu sind sie allzu vage.



SOULS OF DARKON
Bei SOULS OF DARKON ist das Bildschirmdisplay anders gestaltet worden, als das gemeinhin bei dieser Art von Spielen üblich ist. Die Grafik ist zwar nur zweifarbig, erfüllt ihren Zweck durch eine gute Gestaltung, die Platz für eigene Phantasie läßt, allerdings bestens. Lobenswert ist auch, daß es hier wenigstens noch so etwas wie eine Help-Funktion gibt.

Wenn man sich das Bildschirmphoto von SOULS OF DARKON ansieht, will man gar nicht glauben, daß dieses Programm es wirklich schafft, eine echte Atmosphäre zu erzeugen – mitnichten!

Dadurch, daß zum einem durch die einfachen Bilder die Phantasie angestachelt wird, was auf der anderen Seite durch eine ausgezeichnete textliche Gestaltung unterstützt wird, gelang es den Programmierern von Taskset, ein Spiel zu schreiben, das in den Köpfen der Spieler mehr werden wird als nur ein einfaches Abenteuer. SOULS OF DARKON ist vom Schwierigkeitsgrad etwa als obere Mittelklasse zu bezeichnen. Einem Abenteuer-Anfänger würde ich es nicht empfehlen, dafür ist es zu knifflig. (HS)

Das geheime Tagebuch des Adrian Mole, Alter 13 3/4

Daß neben den absoluten Top-Games auch so mancher Humbug aus England zu uns herüberkommt, ist eine unleugbare Tatsache. Und in diesem Wust aus Unsinn ist auch manches Programm, das für uns Deutsche mangels des kulturellen Hintergrundes unverständlich ist. Das Tagebuch des Adrian Mole ist so ein Fall. Basierend auf einem in England sehr populären Buch, haben sich die Programmierer von Level 9 und Mosaik Software zusammengetan und daraus eine Art Adventure gemacht. Die Handlung gibt einfach und ergreifend den Alltag eines Schülers namens Adrian Mole wieder, der, wie sollte es anders sein, sehr englisch ist.

Die Affären dieses Schülers und dessen Eltern, seine Probleme in der Schule, die ersten Kontakte zu Mädchen, die Rivalität unter den Schülern und einiges mehr bilden die Handlung von Adrian Mole. Im großen und ganzen sind das nun ja auch Dinge, mit denen man sich in »Good old Germany« herumschlagen muß, aber bei uns ist doch alles etwas anders – oder ist Ihnen ein Fall bekannt, in dem ein Schüler einen Verweis für das Tragen von roten Strümpfen in der Schule bekam? In England, mit seiner sehr strengen Schulkleiderordnung, ist dies nicht erlaubt, und als Adrian in Ermangelung anderer Socken eben einmal rote anzieht, bekommt er dafür auch sogleich einen strengen Verweis. Aus diesem Sachverhalt entwickelt sich ein regelrechter Konflikt zwischen Schülern und Lehrern, als nämlich alle anderen aus Solidarität zu Adrian ebenfalls rote Socken anziehen. Dieses kleine Beispiel sollte Ihnen zeigen, daß vieles in diesem Programm doch sehr englisch ist und jemandem, der außer der Sprache nicht auch noch ein wenig über die englische Lebensart Bescheid weiß, unverständlich bleibt. Abgesehen davon ist Adrian Mole kein Abenteuerspiel im herkömmlichen Sinne, sondern eigentlich eine illustrierte Novelle, bei der man ab und zu die Möglichkeit hat, in das Geschehen einzugreifen. Praktisch sieht das so aus: Der Text des Tagebuches scrollt auf Tastendruck jeweils um ein Textfenster nach oben, nur gelegentlich bekommt man die Möglichkeit, für Adrian zwischen drei Alternativen zu wählen. Ein Beispiel: Wenn Adrian in der Schule das Sockenproblem zu meistern hat, können Sie für ihn zwischen drei Alternativen wählen:

1. Weiterhin rote Socken tragen.
2. Einen Kompromiß schließen und einen roten und einen schwarzen anziehen.
3. Klein beigeben und schwarze Socken tragen.

Durch Ihre Wahl verändern Sie einen Score, der in Prozent anzeigt, wie positiv oder negativ Sie mit Ihren Entscheidungen das Leben des Adrian Mole beeinflussen. Allerdings ändert sich bei einer Wahl nur der darauffolgende Score, nicht aber das weitere Spiel. Kurz gesagt, es ist eigentlich ein Roman auf dem Computer, der dem Spieler suggeriert, daß er Einfluß auf den Ablauf habe, was aber effektiv Unsinn ist. Die Geschichte bleibt gleich. Für Leute, die sich einmal im Alltag eines englischen Schülers umsehen möchten und über gute Englischkenntnisse verfügen, wäre dieses Programm bisweilen interessant. Aber Spieler oder Adventurecracks werden sich gewiß ganz schön ärgern, und sagen: "Hätt ich bloß nicht..."

(HS)

Gamers Message

Doomark der Finstere überzieht seit einiger Zeit auch deutsche Schneider Rechner mit seinem Eis-Atem. Daß "Lords of Midnight" nicht allen Spielern einleuchtet, ist bei der Komplexität des Programmes gewissermaßen vorprogrammiert. Deshalb finden Sie hier nun einige nützliche Tips, um Doomark in seine Schranken zu weisen.
Schicken Sie Morkin nach Norden und Luxor nach Osten. Doomark wird die Wärme in Luxors Ring fühlen und die meisten seiner Schergen hinter ihm hersenden. So kommt Morkin relativ unbeschadet voran.

Die Eiskrone befindet sich im Turm des Bösen. Sie werden entweder Farflame den Drachenlord, oder Lorgrim den Weisen brauchen, um die Krone zu zerstören. Sollte es Ihnen nicht gelingen, einen der beiden zu rekrutieren, so besteht noch die Möglichkeit, sie in den Spiegelsee zu werfen, um sie unschädlich zu machen.

Warten Sie in den Todesbergen bis die Nacht hereinbricht. Wenn dann der Morgen dämmert, gehen Sie schnell in den Turm, nehmen die Krone und verkrümeln sich wieder in die Berge.

Diese Tips gelten, wenn Sie versuchen wollen, der Bedrohung auf dem Abenteuerweg Herr zu werden. Liegt Ihnen allerdings die Kriegskunst mehr am Herzen, und Sie wollen versuchen, das Böse mittels Waffengewalt zu vertreiben, lesen Sie die nun folgenden Tips. Sie werden Ihnen sicherlich gute Dienste leisten.

Lassen Sie Morkin nach Norden reisen, um die Aufmerksamkeit der mit der ICE FEAR heraufbeschworenen Dämonen auf ihn zu lenken. So können Sie mit Ihren anderen Heerführern ungestörter agieren.

Wenn einer Ihrer Heerführer keine Armeen mehr hat, vermeiden Sie jede kriegerische Auseinandersetzung.

Rekrutieren Sie so viele Heerführer wie möglich. Doomark ist mächtig, und Sie werden alle Hilfe brauchen, die Sie kriegen können.

Lassen Sie Morkin keine andere Wahl, als Umherzureisen, in Städten zu schlafen, und die Ice-Fear zu bekämpfen, da er sowieso nicht in der Lage ist, sonderlich viele Krieger zu rekrutieren.

Allerdings ist Morkin trotz allem in der Lage, Farflame, den Drachenlord, zu rekrutieren. Dieser ist in Schlachten ein mächtiger Mitstreiter, und kann an einem Tag halb Midnight überfliegen.

Den Schattenlord finden Sie, wenn Sie vom Anfangspunkt des Spieles N N W gehen.

Corleth kann in der Zitadelle von Gard den Lord von Gard rekrutieren, indem er ihm den Pokal der Träume übergibt.
Soweit die Tips zu Midnight.

Denjenigen, die bei dem ersten Teil des Data-Media-Abenteuers MARCO POLO festsitzen, wird das Folgende sicherlich gut gefallen: Der komplette Lösungsweg, wie immer in solchen Fällen auf den Kopf gestellt. Bedanken möchte ich mich hierfür bei Axel Wesseling, der ihn uns freundlicherweise zur Verfügung gestellt hat. hat.

N / ZEIGE PASS / SAGE 'GIB GOLD'/ SIE / NIMM BRIEF / S / W / KAUFE BROT / 99 / O / O / KAUFE WEIN / SAGE 'KOMM MIT' / ? / ? / W / S / W / GEHE SCHIFF / ZEIGE BRIEF / SAGE 'FAHR LOS' / W / S / SCHAUE SCHIFF / TÖTE PIRATEN / NIMM GOLD (Mehrmals) / O / O / S / NIMM SCHLÜSSEL / KAUFE BROT / N / O / S / S / S / NIMM TROMPETE / KAUFE BROT / W / BLASE TROMPETE / GIB GOLD / GIB ZEHN GOLDSTÜCKE / W / W / N / W / SCHAUE KAKTUS / NIMM SAPHIR / O / S / O / S / O / RETTE MÄDCHEN / SAPHIR / O / O / S / SCHAUE HAUS / ZEIGE MÄDCHEN.

Und für diesen Monat war das dann auch schon wieder alles. Ich hoffe, daß ich mit diesen Tips dem einen oder anderen ein wenig helfen konnte und wünsche allen Abenteurern und passionierten Computerspielern einen schönen regenarmen Sommer. Bis nächsten Monat.

Euer
HEINRICH

P.S.: Kleine Überraschungen können Sie erleben, wenn Sie bei Spindizzy im Titelbild die Tasten "F" und "X" gleichzeitig drücken oder bei "Thing on a Spring" im Titelbild die Tasten "THING.

Inserentenverzeichnis Nr. 6



„Schneider CPC 7/86" erhalten Sie ab 26. Juni bei Ihrem Zeitschriftenhändler


Schneiderware bringt interessante Erweiterungen für alle CPC-Rechner. u.a. haben wir eine komplette Centronics-Schnittstelle zum Selbstbau vorbereitet.



CPC International präsentiert die große Spiele-Ecke. Mit allem was dazugehört: neue Software, Joysticktests und viele interessante Dinge aus dem Spiele-Sektor. Sie dürfen gespannt sein!


Wir testeten u.a. die seit kurzem erhältliche Centronics/serielle-Schnittstelle für den Joyce PC. Die Schnittstelle zur Außenwelt – was sie verspricht und was sie hält.

Berichte

Programmieren- aber richtig!
– unser Bericht gibt Auskunft über die verschiedenen Möglichkeiten der Basic-Programmierung. Welche Überlegungen zu einer ordentlichen Programmplanung gehören und wie die Umsetzung in die Praxis vorgenommen wird, sind Bestandteile dieses Artikels.

Bits und Bytes
– Basic und Maschinencode – welche Möglichkeiten gibt es zur Ablage? Wenn man nicht nur in Basic, sondern auch in compilernahen Programmiersprachen wie Assembler, Forth etc. programmiert, dann stellt sich oft die Frage, wo man diese compilierten Maschinenprogramme am besten speichert. Wir zeigen Ihnen den besten Weg.

Der große Spiele-Teil
– CPC International hat für alle Freunde des Computerspiels einen wahren Leckerbissen vorbereitet. Unser großer Spiele-Teil zeigt Ihnen u.a. die top-aktuelle Software und gibt viele Tips und Lösungshilfen. Des weiteren haben wir viele Joysticks getestet, die wir Ihnen in Wort und Bild vorstellen werden.

Tips & Tricks

Neue Mathefunktionen
– dieses tolle Hilfsprogramm stellt insgesamt 24 neue mathematische Funktionen zur Verfügung. Damit wird das Rechnen zum Kinderspiel!

Headerlose Kassettenoperationen
– um Programme mit dem Kassettenrekorder schneller laden und abspeichern zu können, hat man entweder die Möglichkeit des Erhöhens der Baudrate oder die des headerlosen Abspeicherns. Diese interessante Variante werden wir Ihnen ausführlich erläutern.

Joyce

Unsere Joyce Rubrik hat erneut Interessantes für Sie parat.
Das Programm "Kalkül" ist dabei der absolute Höhepunkt, denn diese Tabellenkalkulation ist wirklich als professionell zu bezeichnen.
Natürlich gibt es wieder viele Tips zum Umgang mit dem Joyce, sowie Hinweise zu Mallard Basic und CP/M.

Serie

Neu: Das Software-Experiment
Unser Autor Matthias Uphoff wird Sie mit einem neuen Projekt bekannt machen. Das Software-Experiment zeigt sehr eindrucksvoll die Grundzüge der künstlichen Intelligenz, aufbereitet mit viel Humor und der Bitte an unsere Leser, aktiv an der weiteren Gestaltung dieser exclusiven Top-Serie mitzuarbeiten. Den Auftakt bietet das Programm "Apfelmännchen" – übrigens das schnellste, das jemals auf CPC-Rechnern programmiert wurde!

Hardware:

Schneiderware
– nach dem Auftakt geht es weiter in das "Eingemachte". Schneiderware bringt im nächsten Monat u.a. eine komplette 8-Bit Centronics-Schnittstelle für alle CPC-Rechner.
Dazu gibt es natürlich die entsprechenden Grundlagenartikel, so daß am Ende nicht nur das fertige Produkt entsteht, sondern auch die Frage "Wie funktioniert das?" beantwortet sein wird.

Software Reviews

Bomb Jack
Eden Blues
Nightshade
Workwriter
Mathepack

CAL

CAL – der Vokabeltrainer
– unsere neue Rubrik "Lernen mit dem Computer" bringt zum Auftakt den Vokabeltrainer. Ein ausgeklügeltes Programm, das die eingegebenen Wortfolgen systematisch abfragt.

CP/M

Wir basteln ein Limbo
– mit Submit-Dateien kann man ohne viel Mühe neue CP/M-Befehle einführen. Zum Beispiel, um gelöschte Dateien bequem wieder "zum Leben zu erwecken".

Bitte Club-Information auf Seite 117 beachten!

Druckerplattform

bestehend aus einer Trägerplatte für den Drucker und einer darunter angeordneten Platte für ca. 1000 Blatt Papier. Beide Platten sind durch stabile Seitenwände miteinander verbunden.
Ansprechendes Design und hochwertiges Material (Acrylglas) zeichnen dieses Produkt aus.
Außenmaß 420 x 400 x 130 mm **79,— DM** **99,— DM**



Die Maus für Schneider

Was Apple und Atari recht ist, ist Schneider billig.

Die Maus ist eine leicht zu bedienende Vorrichtung, die mit der Hand geführt wird. Die Bewegungen der Maus werden auf ein Zeichen (z.B. einen Pfeil) auf dem Bildschirm Ihres Computers übertragen. Anstatt komplizierte Befehle einzutippen, zeigen Sie mit der Maus einfach auf die gewünschte Funktion und drücken eine der beiden Tasten.
In Verbindung mit einem Graphik-Paket verfügen Sie über ein Werkzeug, welches Ihren Bildschirm in die Leinwand eines Künstlers verwandelt. Es ermöglicht Ihnen, Gemälde zu erstellen ohne die Verwendung von Papier und Zeichengerät.





Ohne Abbildung:
Schneider-Monitor-Verlängerungskabel
für CPC 464
Best.-Nr. 826 — **19,50 DM** / **14,90 DM**
für CPC 664 + 6128
Best.-Nr. 827 — **39,– DM** / **29,– DM**

Staubschutzhaube

f. Grün-Monitor	816	28,—	37,50
f. Farb-Monitor	817	28,—	37,50
f. Drucker	818	16,90	22,50
f. CPC 464	819	16,90	22,50
f. Floppy	820	14,90	19,90
f. 664	–	16,90	22,50
f. 6128	–	16,90	22,50

Schneider-Maus

Wird über einen Spezial-Adapter an den Joystickport von Schneider CPC 464, CPC 664 oder CPC 6128 angeschlossen.
Adapter und Unterprogramme zum Einbau der Maus in bestehende BASIC-Programme, sowie Graphik-Paket mit Maus-Unterstützung im Lieferumfang enthalten.
Deutsche Anleitung; mit Software auf 3"-Diskette **169,— DM** **199,— DM**

Schwarze Preise: Verkaufspreise für Nicht-Clubmitglieder
Rote Preise: nur für Clubmitglieder

Datenrecorder MC 3810

für CPC 664/6128

anschlußfertig inkl. Kabel und Netzteil.
Best.-Nr. 836 **59,– DM** **78,– DM**

MULTIDATEI (Data Media)

Ein universelles Dateiverwaltungsprogramm.
Cass. Best.-Nr. 141 — **89,– DM** **49,– DM**
Disk. 3" Best.-Nr. 241 — **99,– DM** **59,– DM**

ROUTINEN, TIPS UND TRICKS (Data Media)

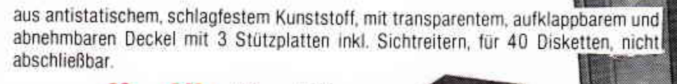
Ca. 50 wichtige Programmroutinen für jeden Programmierer, wie z.B. Sortierroutine, Inkeyroutine, Diskettenroutine usw.
Cass. Best.-Nr. 151 — **79,– DM** **39,– DM**
Disk. 3" Best.-Nr. 251 — **89,– DM** **49,– DM**

GREDI (Data Media)

Hilfsprogramm zur Herstellung von Hires-Grafiken.
Cass. Best.-Nr. 142 — **49,– DM** **29,– DM**
Disk. 3" Best.-Nr. 242 — **59,– DM** **39,– DM**

NEU Diskettenbox für 3"-Disketten

aus antistatischem, schlagfestem Kunststoff, mit transparentem, aufklappbarem und abnehmbaren Deckel mit 3 Stützplatten inkl. Sichtreitern, für 40 Disketten, nicht abschließbar.
29,— DM **44,— DM**
dito mit Schloß **45,— DM** **54,— DM**



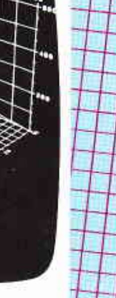
FOCUS (Data Media)

Programm zur perspektivischen Darstellung von 3D-Grafiken. Stellen Sie z.B. ein Haus dar und verändern Sie durch Eingabe der Winkel die Perspektive!

Cassette Best.-Nr. 138
79,– DM **49,– DM**

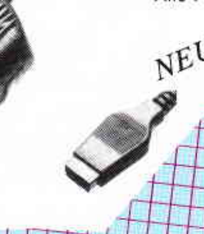
Diskette 3" Best.-Nr. 238
89,– DM
59,– DM













Speedking

Joystick der Luxusklasse, der Hand angepaßt!
Alle Funktionen über Mikroschalter gesteuert.
NEU
49,— DM **59,— DM**





Data Media Mailorder und Computerclub

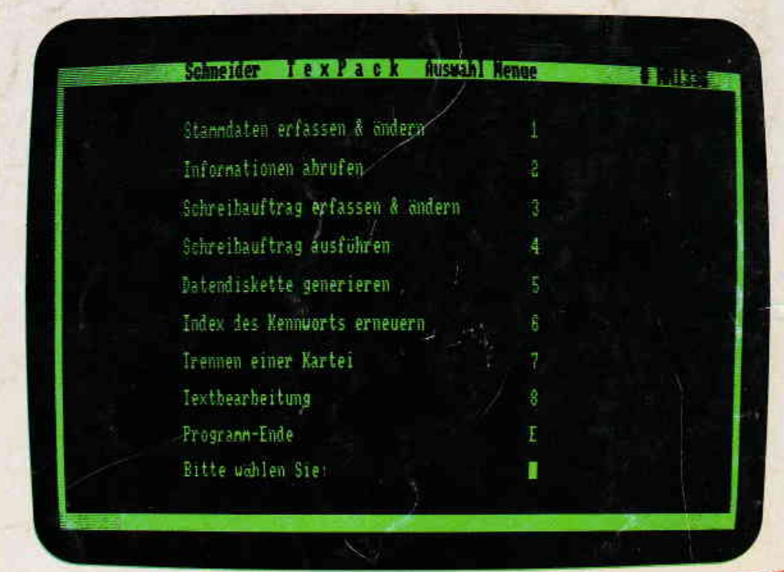
Bestellen Sie noch heute, Lieferung erfolgt umgehend! Zahlung per Vorkasse oder Nachnahme zuzügl. Porto- bzw. Nachnahmegebühr (Nachnahme ins Ausland ist nicht möglich).
Weitere Artikel in unserem Club- oder Mailorderkatalog. Bitte anfordern (2,– DM für Rückporto beilegen).
Data Media GmbH, Ruhralle 55, 4600 Dortmund, Tel.: (02 31) 12 50 71-3

Bestellkarte benutzen!

»TexPack«

Die professionelle Textverarbeitung von Schneider für den »CPC«

Diese professionelle Textverarbeitung auf Diskette, kann durch ihre Vielzahl von Möglichkeiten für verschiedenste Zwecke eingesetzt werden.
Zum Programmpaket »TexPack« gehört neben der eigentlichen Textverarbeitung auch eine komfortable Adreß- und Dateiverwaltung.



Die Leistungsmerkmale des »TexPack« in Stichworten:

- Zeichen einfügen und löschen
- Zeilen einfügen und löschen
- Absätze einfügen und löschen
- Begriffe suchen und ersetzen
- Wort löschen, Zeilenrest löschen
- Zeilen aufbrechen und anschließen
- Bausteinverarbeitung
- Adreßbe- und -verarbeitung
- Textbreite bis zu 240 Zeichen pro Zeile
- Fließtexteingabe
- Randausgleich auch zur nachträglichen Änderung der Textbreite
- Block- oder Flattersatz wahlweise
- Freie Wahl des linken Randes für beliebige Textabschnitte

Folgende Druckausgaben sind variierbar:

- Schriftbreite
- Zeilenabstand ein-, eineinhalb- und zweizeilig
- Formelschreibweise (Hoch- und Tiefstellung)
- verschiedene Hervorhebungsarten: Unterstreichen, Fettdruck etc.
- Normalschrift oder Korrespondenz-Qualität (Near-Letter-Quality)

Diskettenprogramm

unverbindliche Preisempfehlung **DM 198.-**

Damit Sie den vollen Leistungsumfang von »TexPack« nutzen können:



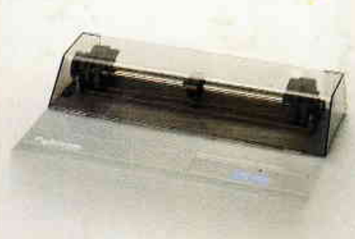
»FD-1« Diskettenlaufwerk als zweites Laufwerk



»CF2« Diskette



Drucker »NLQ 401« (Near Letter Quality)



Traktoraufsatz »SFT 401« zum »NLQ 401«

Unsere Software-Palette wird ständig erweitert. Fragen Sie Ihren Händler nach den aktuellen Neuerscheinungen.

